



WE ARE HIRING! WE ARE HIRING!
WE ARE HIRING! WE ARE HIRING! WE ARE HIRING!
WE ARE HIRING! WE ARE HIRING! WE ARE HIRING!

dais de NBI
GERMAN NETWORK FOR BIOINFORMATICS INFRASTRUCTURE

HUMAN
TECHNOPOLE



CSBDeep - a deep learning toolbox for
microscopists.

What can I do with my data?

Available tools

Questions

Contact

Impressum

CSBDeep - a deep learning toolbox for microscopy image restoration and analysis

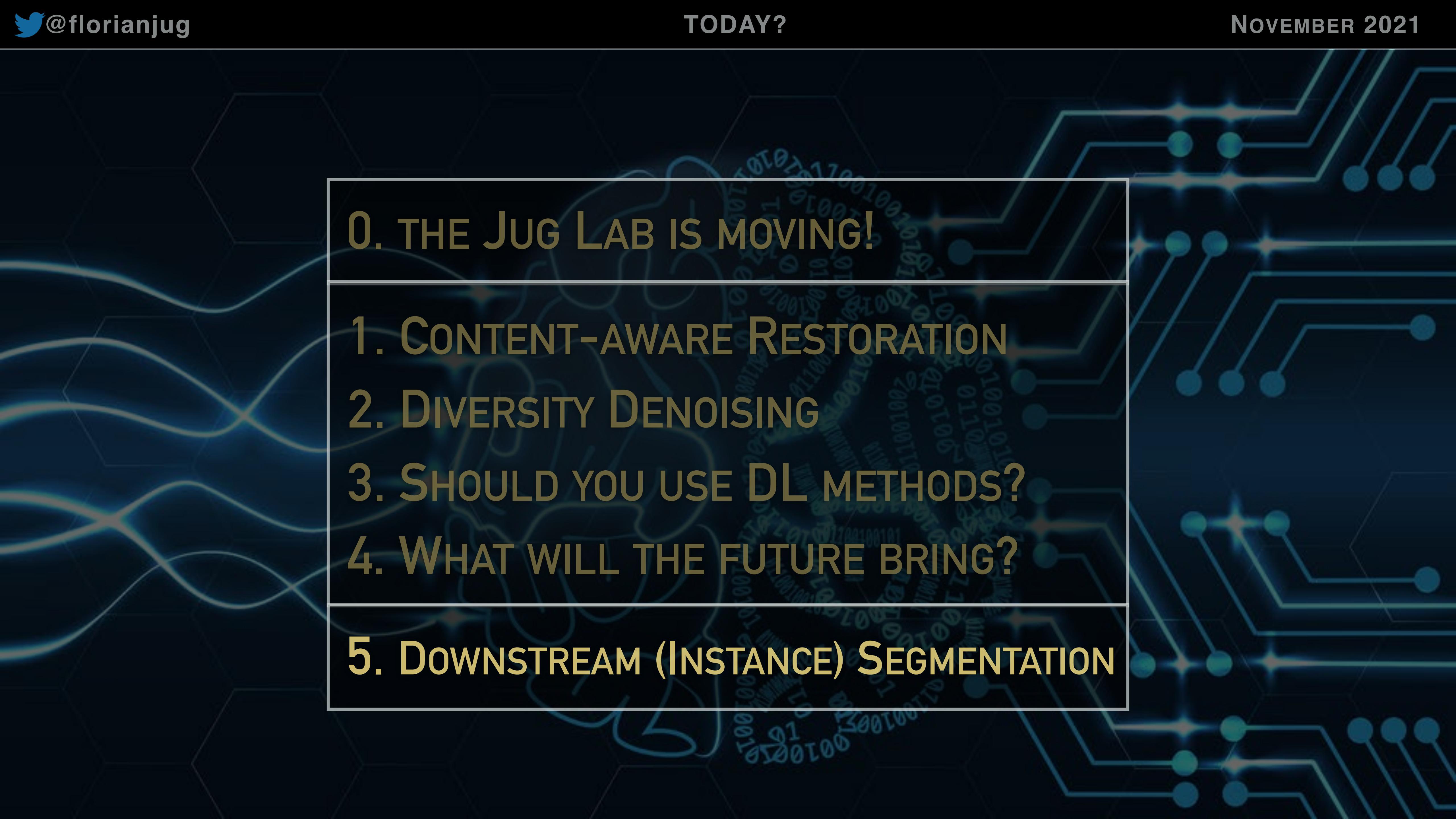
Fluorescence microscopy is a key driver of discoveries in life-sciences, and the CSBDeep toolbox is offering a collection of state-of-the-art methods for content-aware image restoration and segmentation. On this website we showcase a number of [real-world scenarios](#) that might be useful in the context of your own work. Additionally, we will describe the respective [methods and tools](#) we provide and link to further resources wherever we can.

The entire CSBDeep toolbox is fully open source and intended to be used from either Python or [Fiji](#).

<http://csbdeep.bioimagecomputing.com>

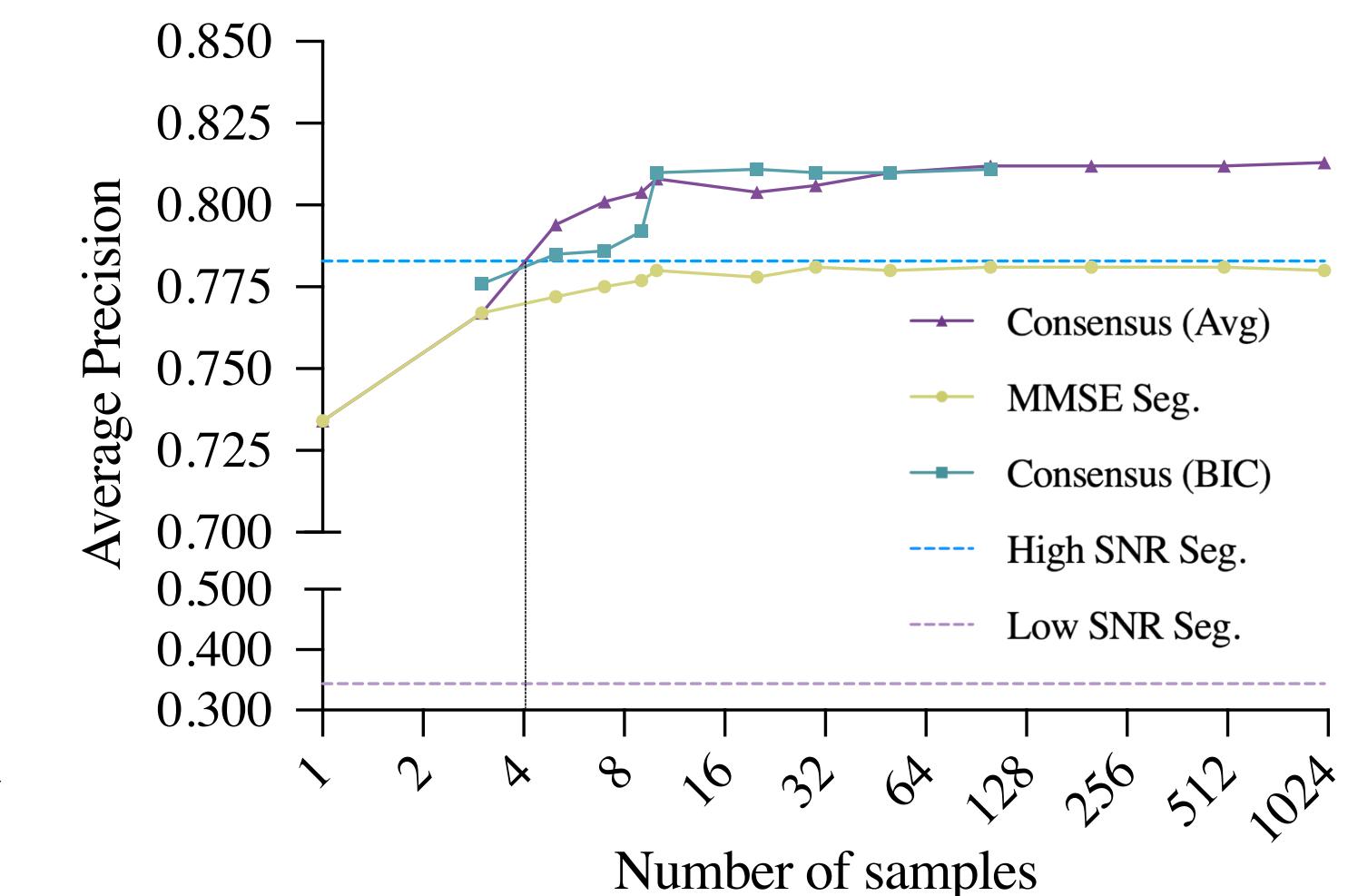
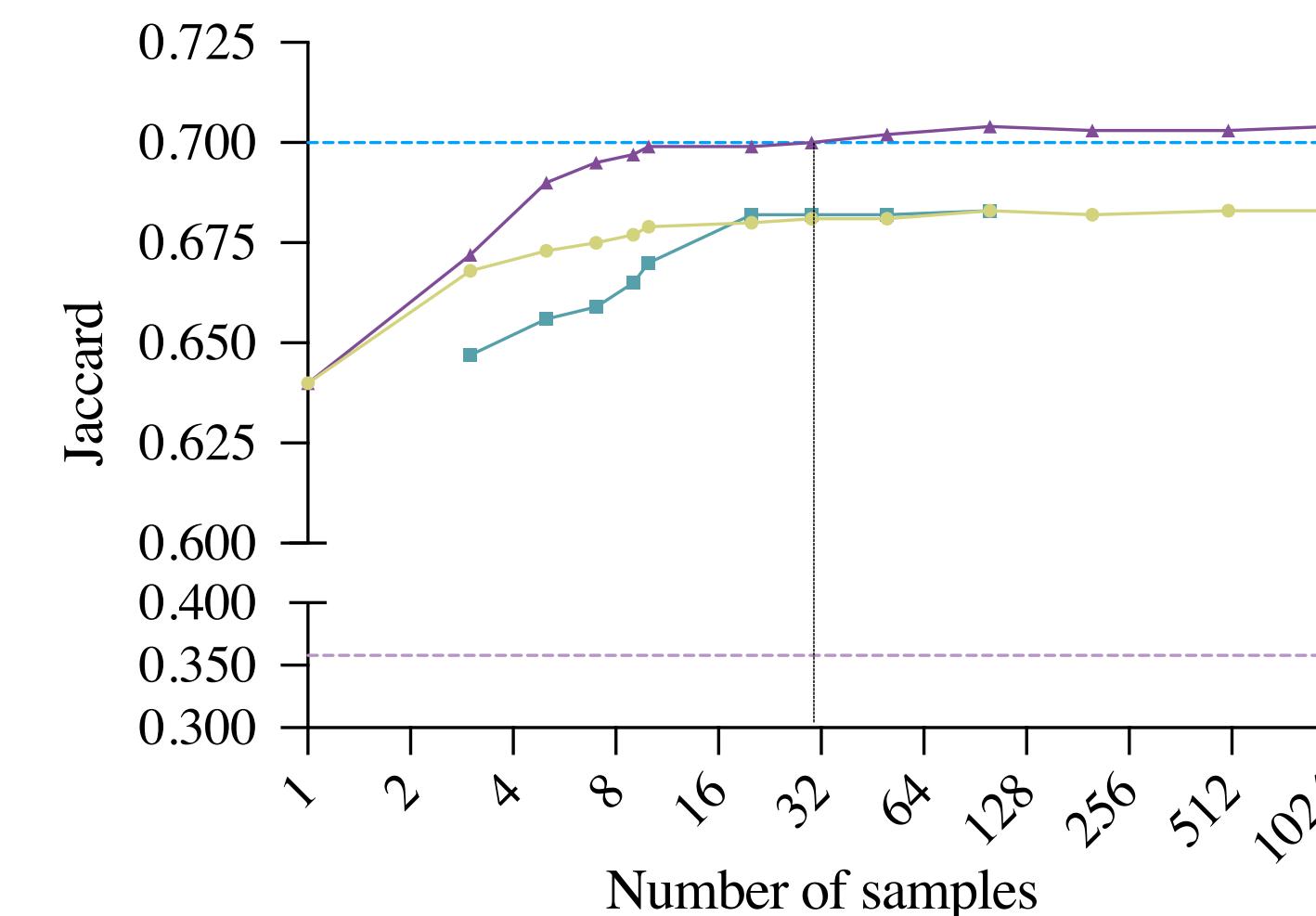
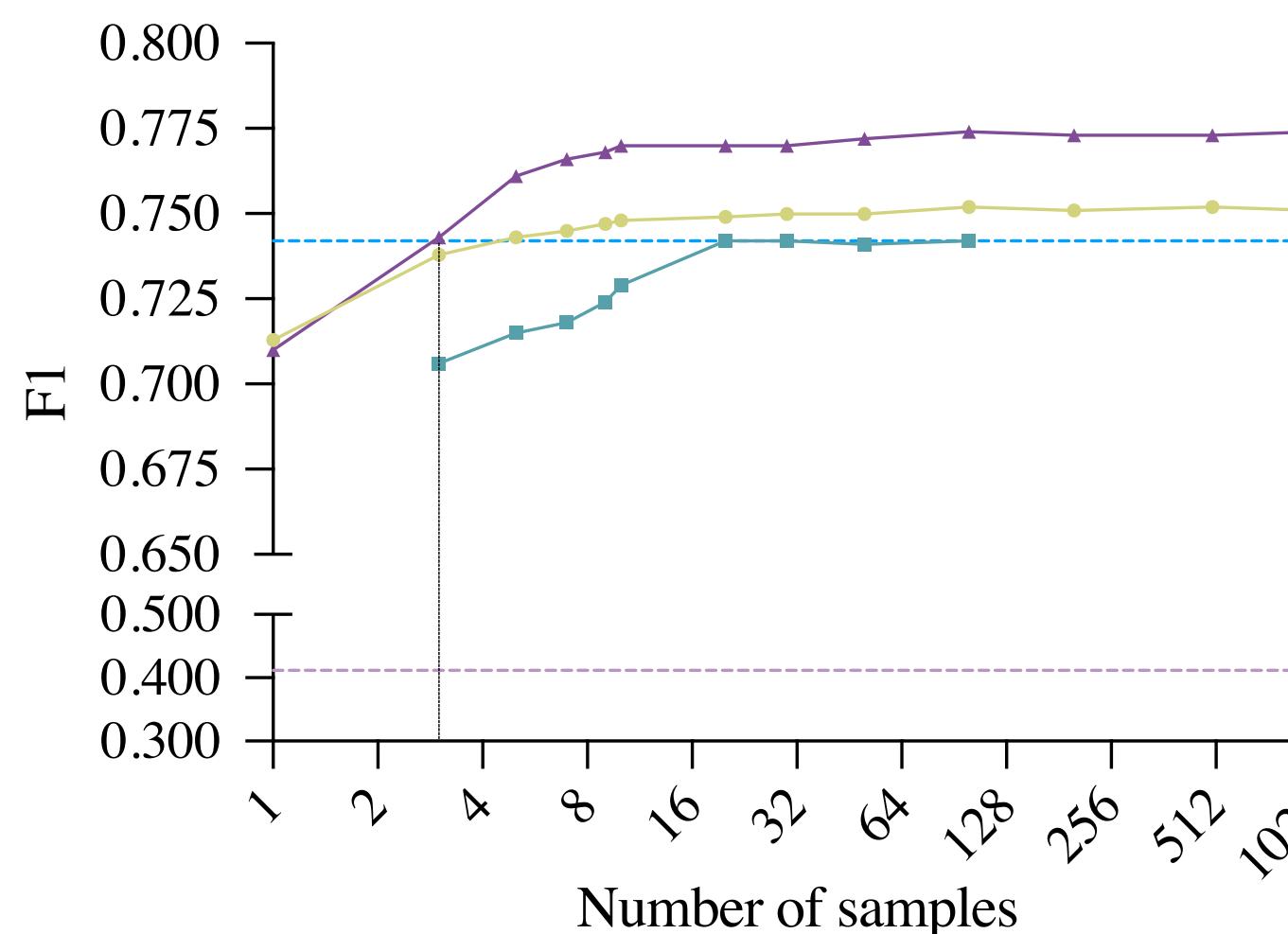
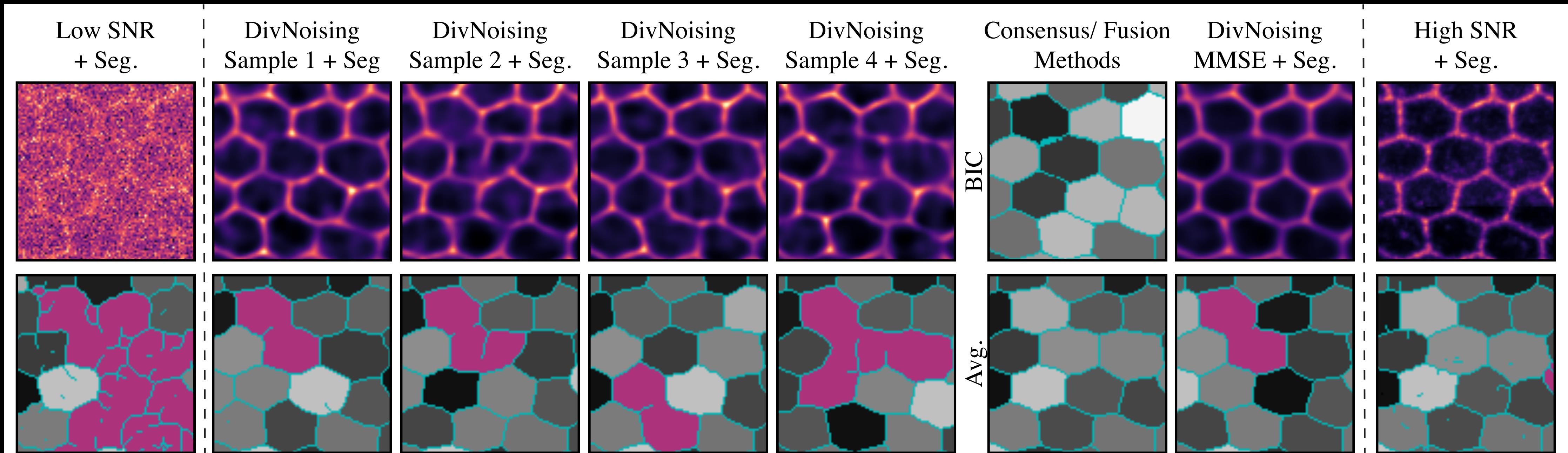


MICROSCOPY IMAGE RESTORATION AND DOWNSTREAM ANALYSIS RECENT IMPROVEMENTS AND (MY) HOPES FOR THE FUTURE

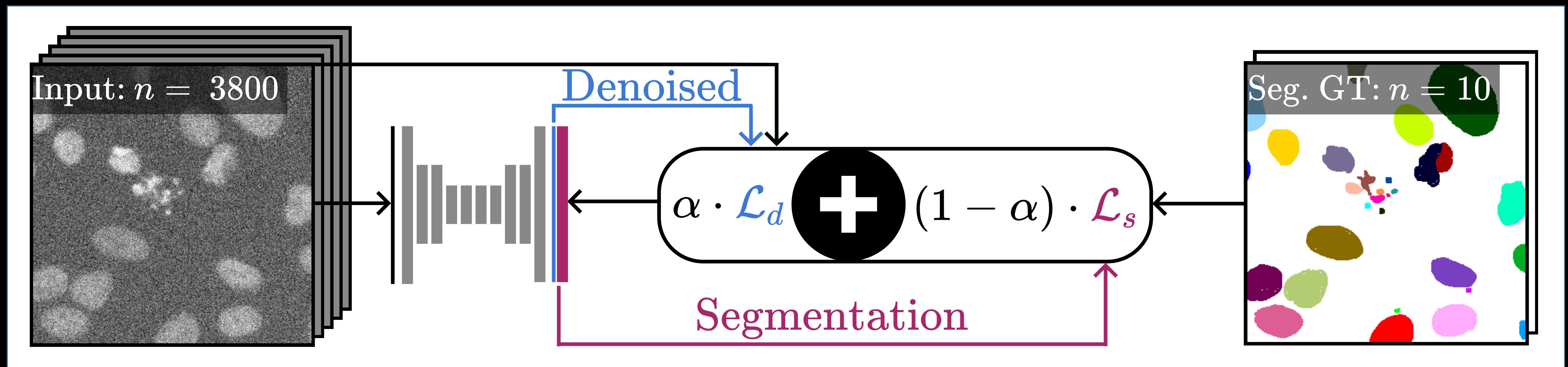
- 
- 0. THE JUG LAB IS MOVING!**
 - 1. CONTENT-AWARE RESTORATION**
 - 2. DIVERSITY DENOISING**
 - 3. SHOULD YOU USE DL METHODS?**
 - 4. WHAT WILL THE FUTURE BRING?**
 - 5. DOWNSTREAM (INSTANCE) SEGMENTATION**

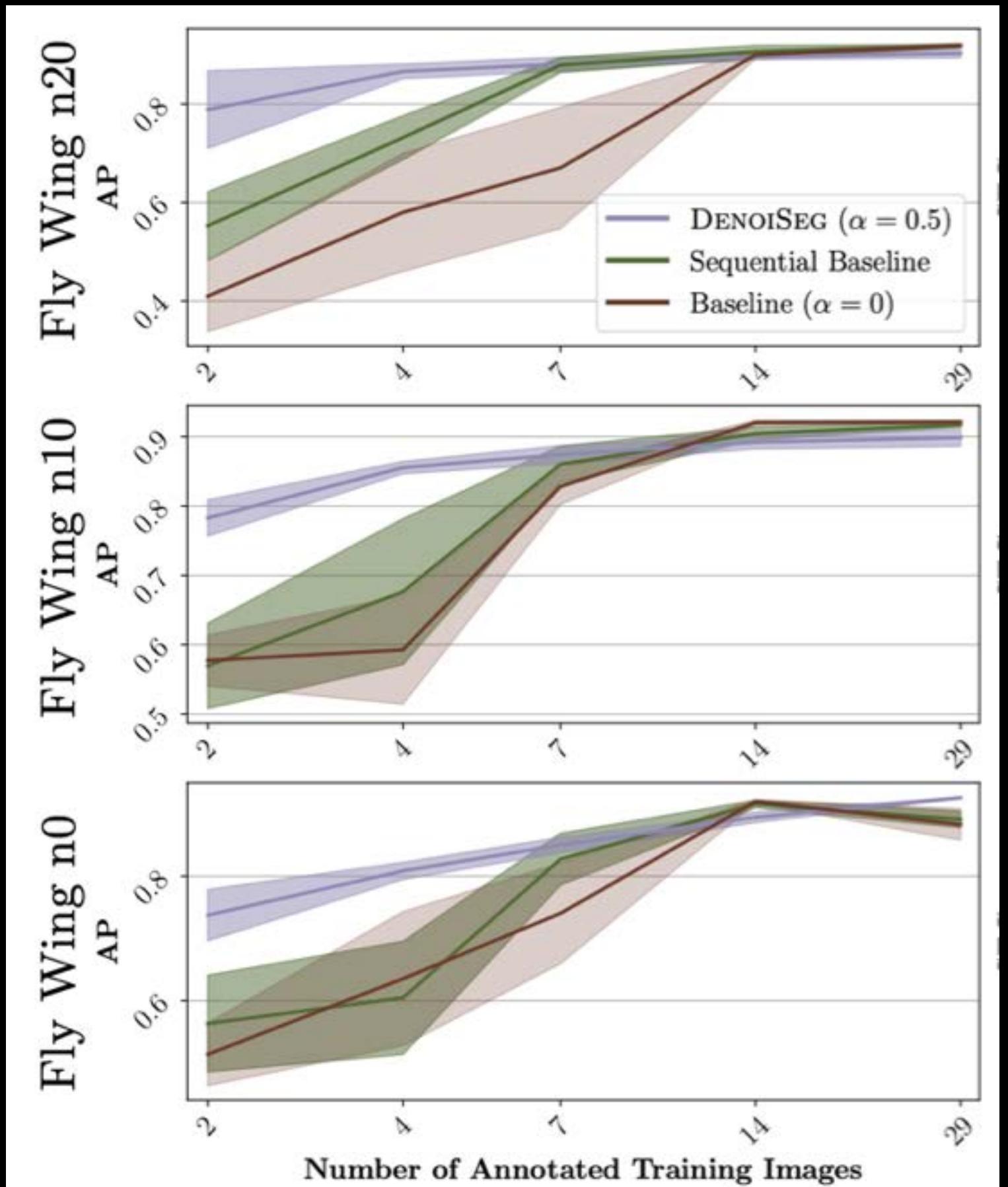
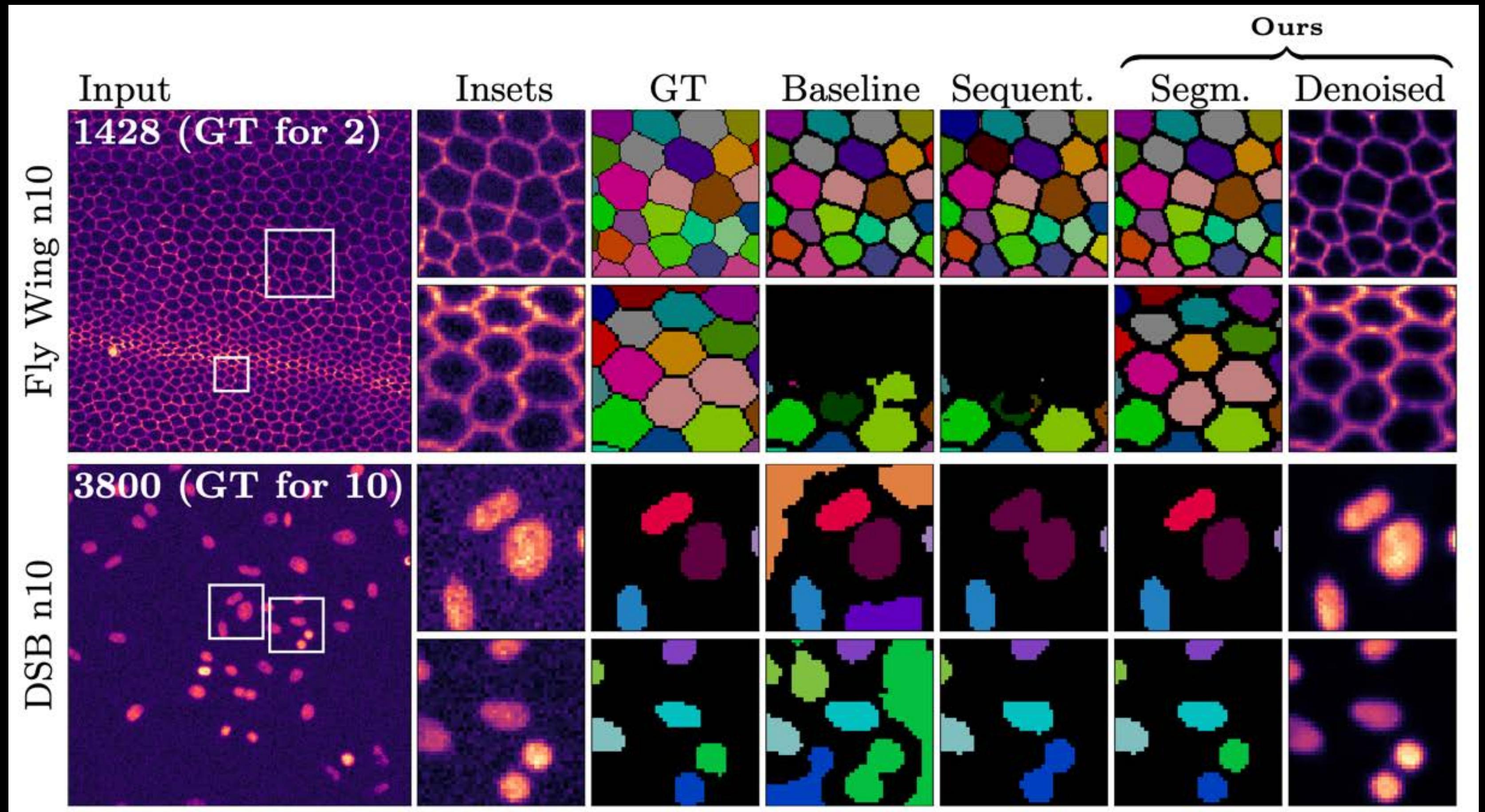
CAN DIVERSITY DENOISING
HELP TO BETTER SEGMENT
OBJECTS IN MICROGRAPHS?

>> Prakash, Krull, Jug 2020 (ICLR)

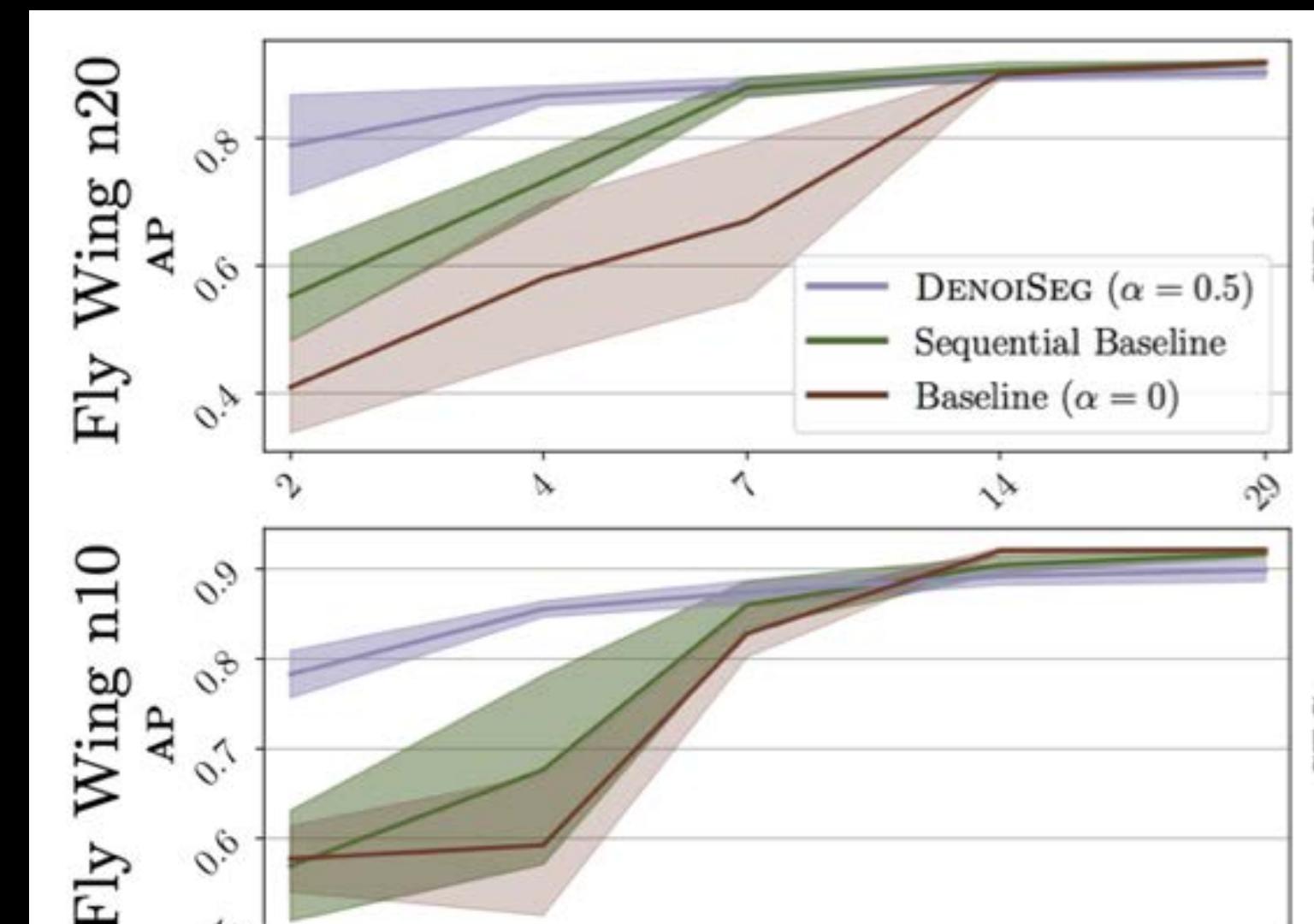
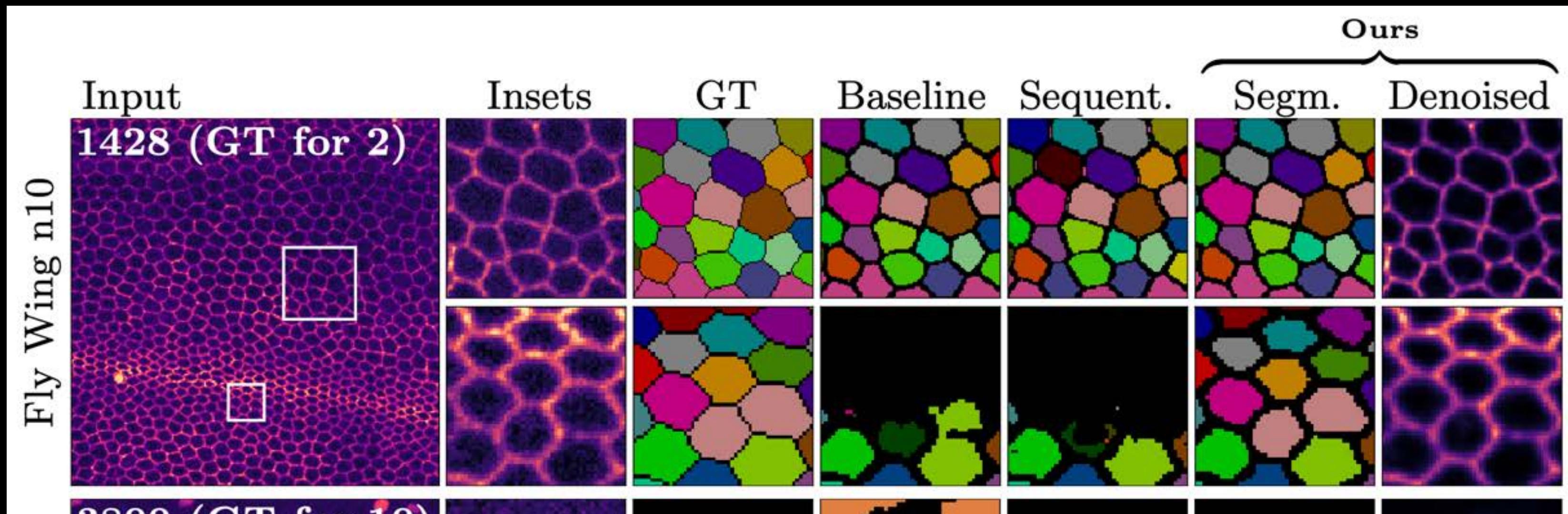
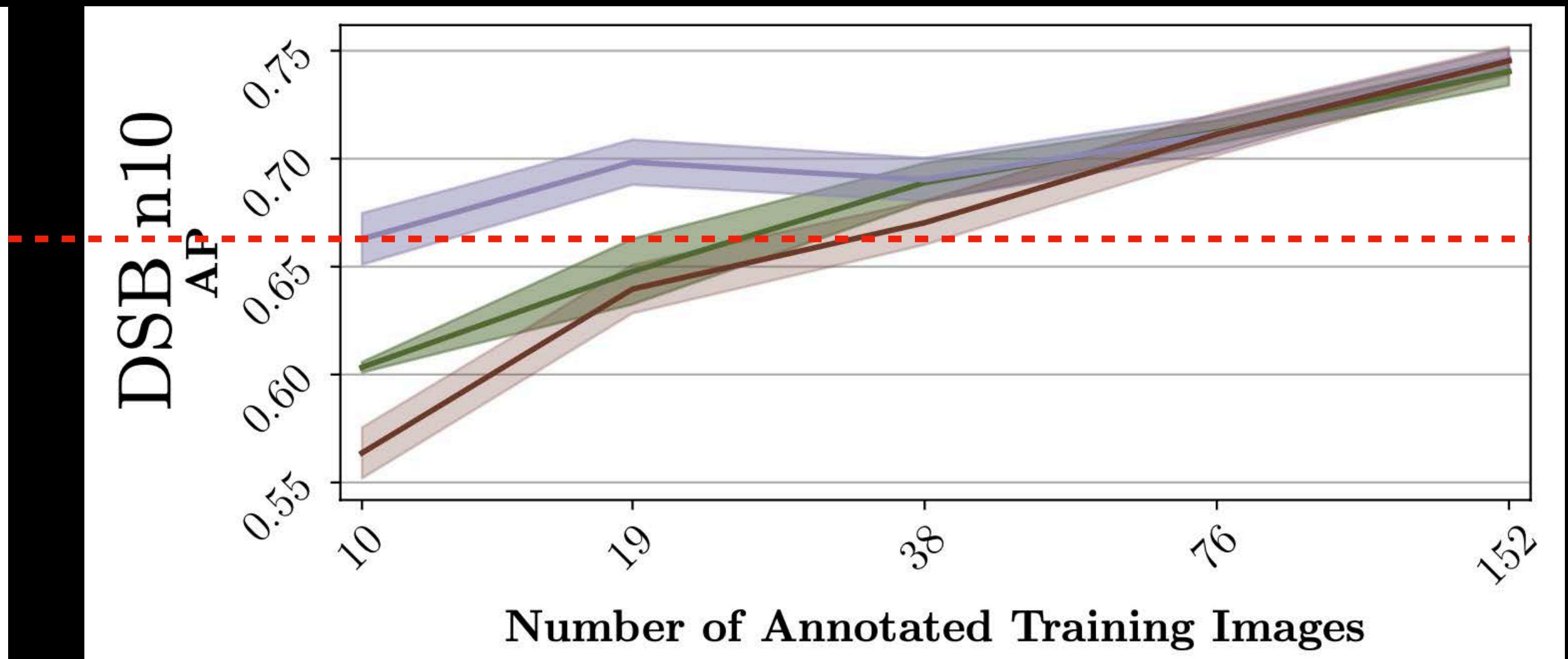
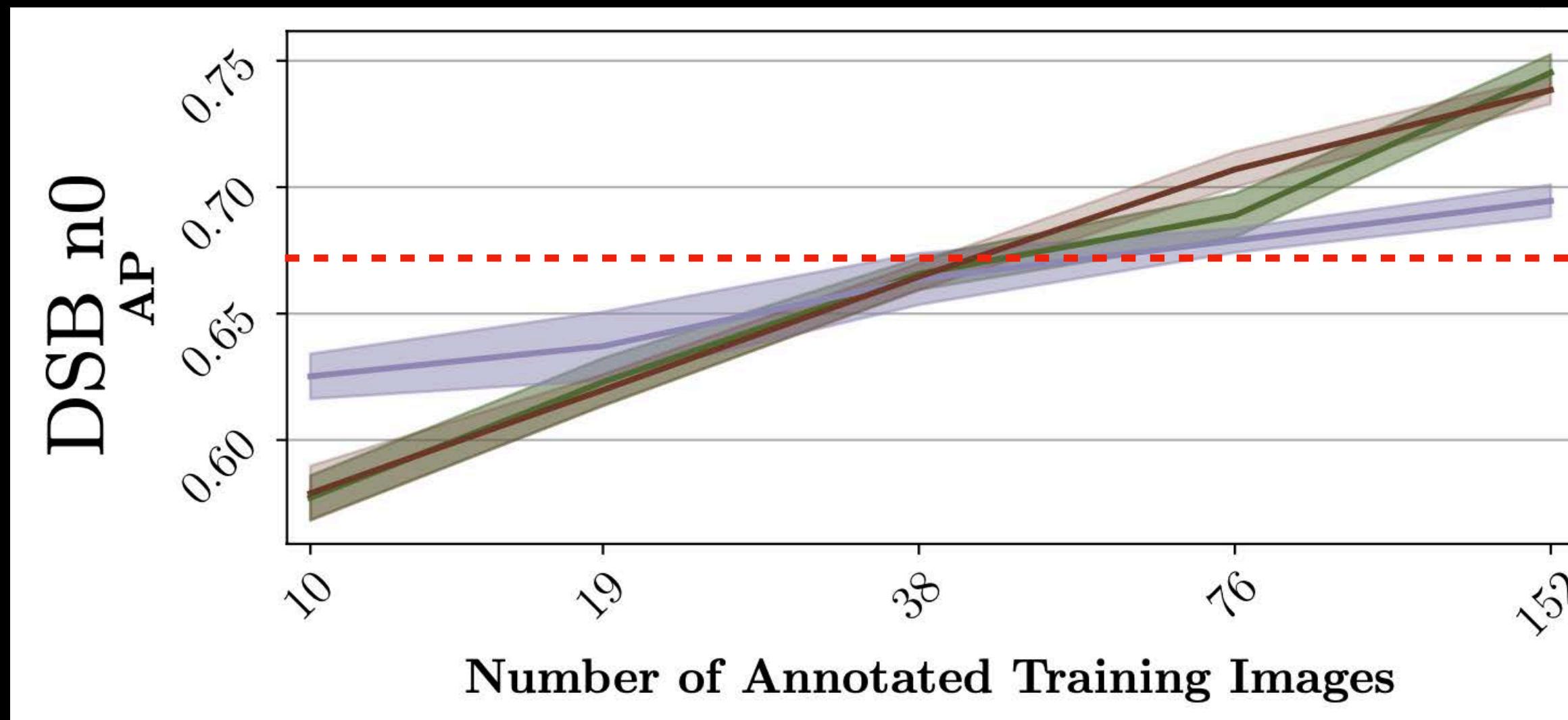


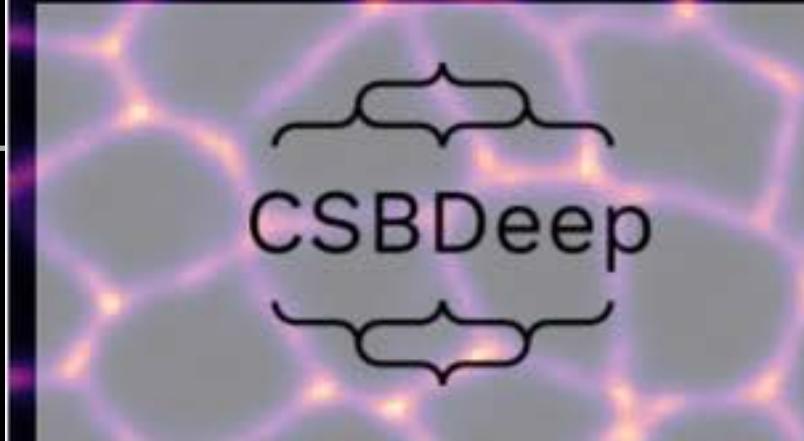
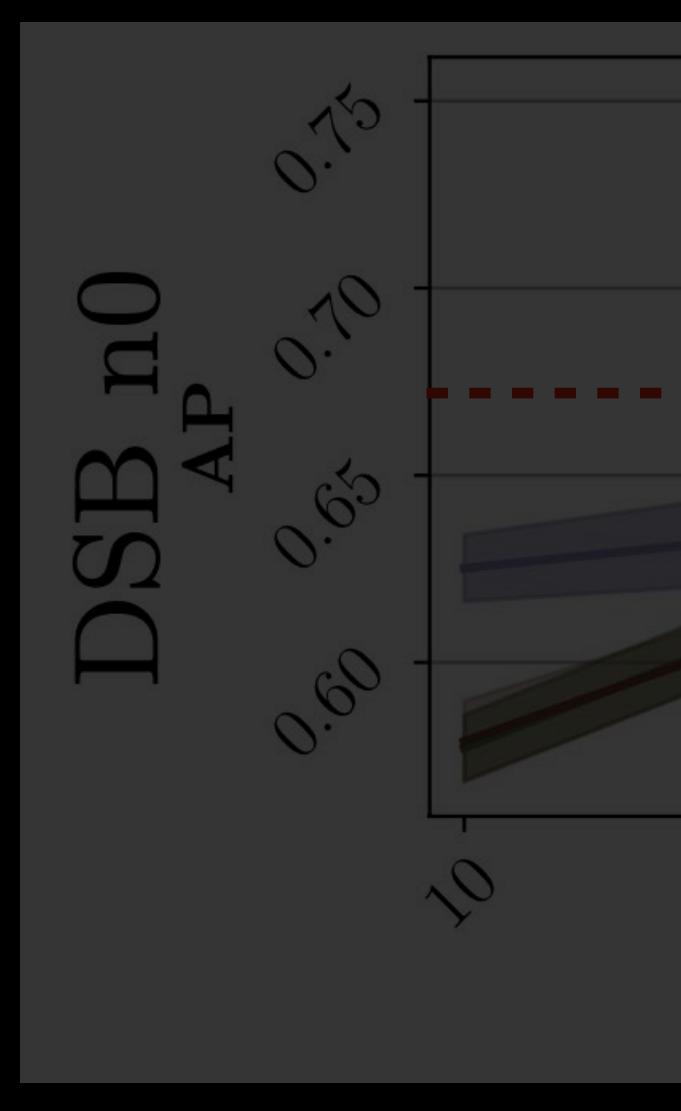
CAN NOISE2VOID HELP TO
IMPROVE SEGMENTATION
PERFORMANCE?





>> Buchholz, Prakash et al. 2020 (BIC@ECCV)





CSBDeep - a deep learning toolbox for microscopists.

What can I do with my data?

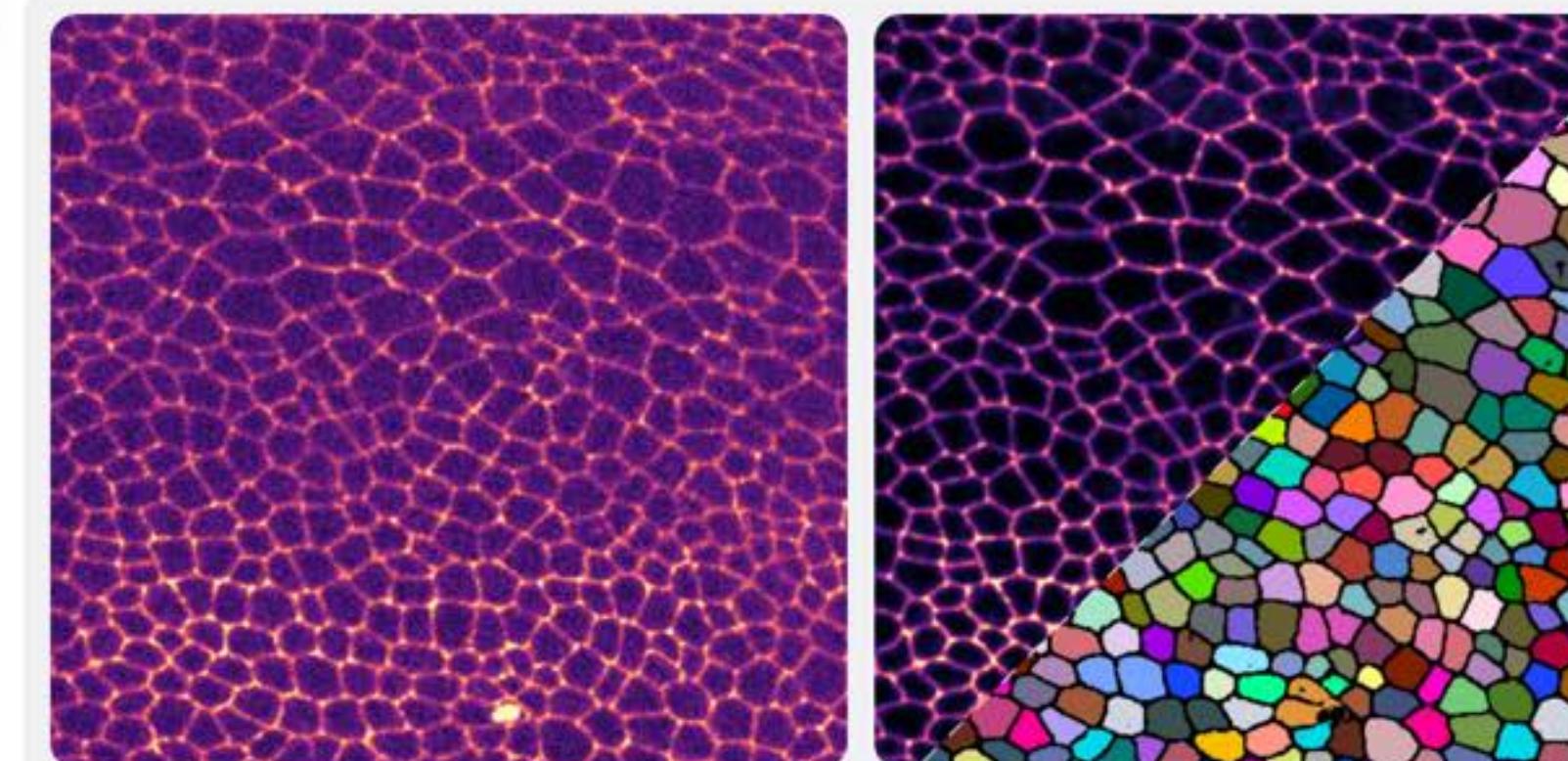
Available tools

Questions

Contact

Impressum

Denoising and Segmentation in 2D (Drosophila wing, e-cadherin)



Used CSBDeep tool:

DenoISeg

If you want to reconstruct this scenario or apply it to your own data, read this page and follow the instructions for DenoISeg (click on this box).

[Inspect exemplary result image](#)

Authorship

This work (method and data) is part of [DenoISeg: Joint Denoising and Segmentation](#) by Tim-Oliver Buchholz and Mangal Prakash et al.

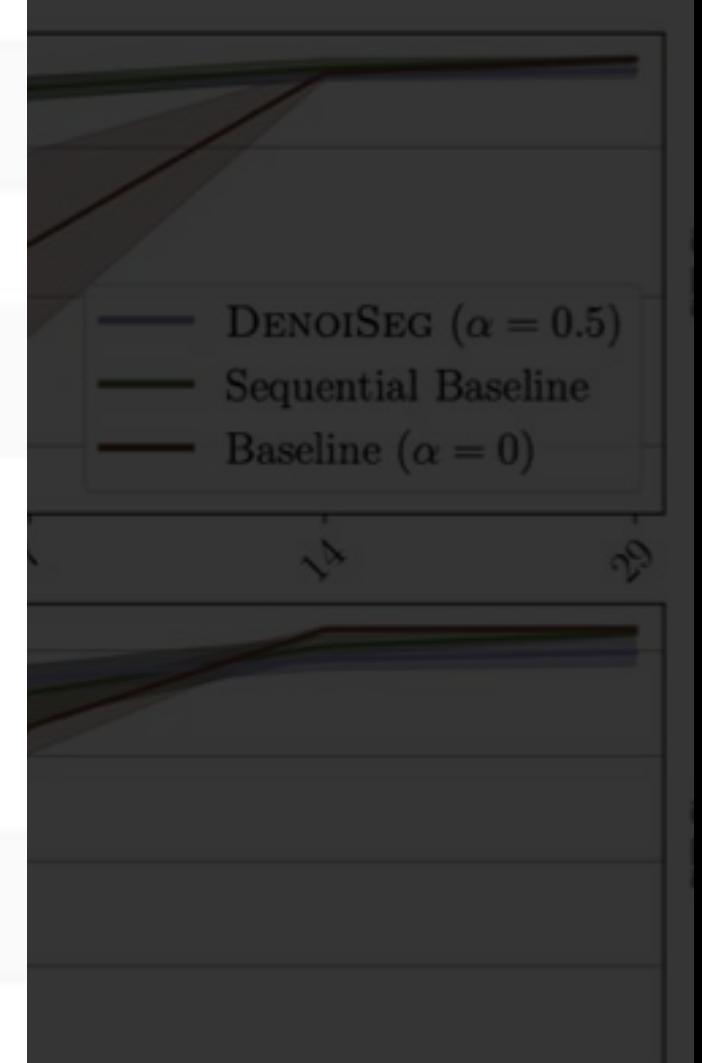
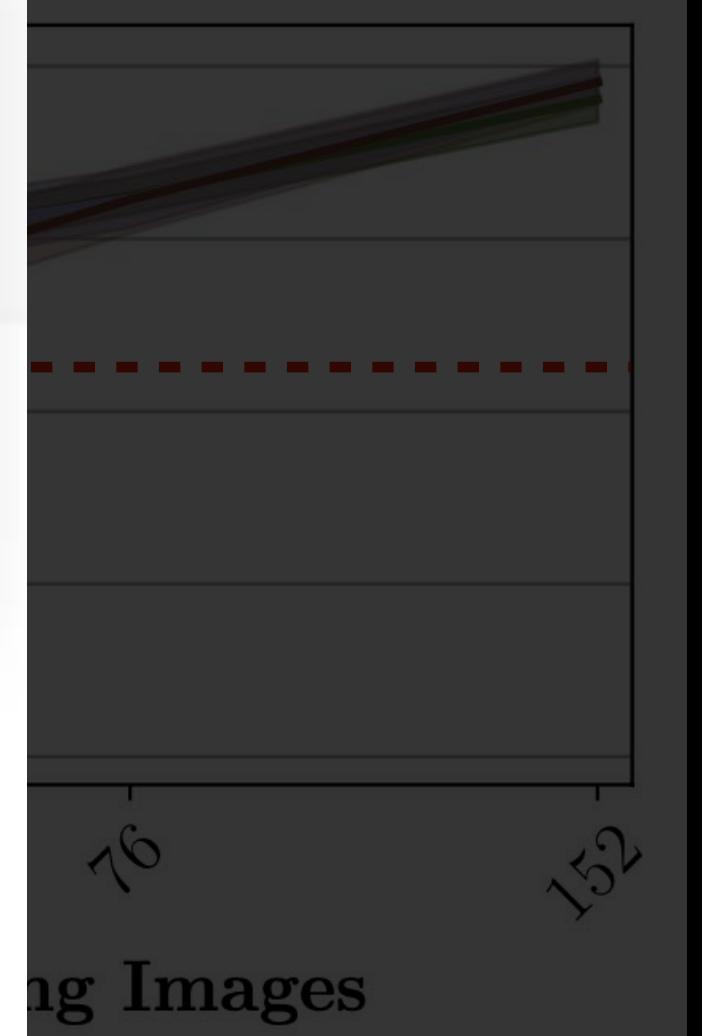
Details

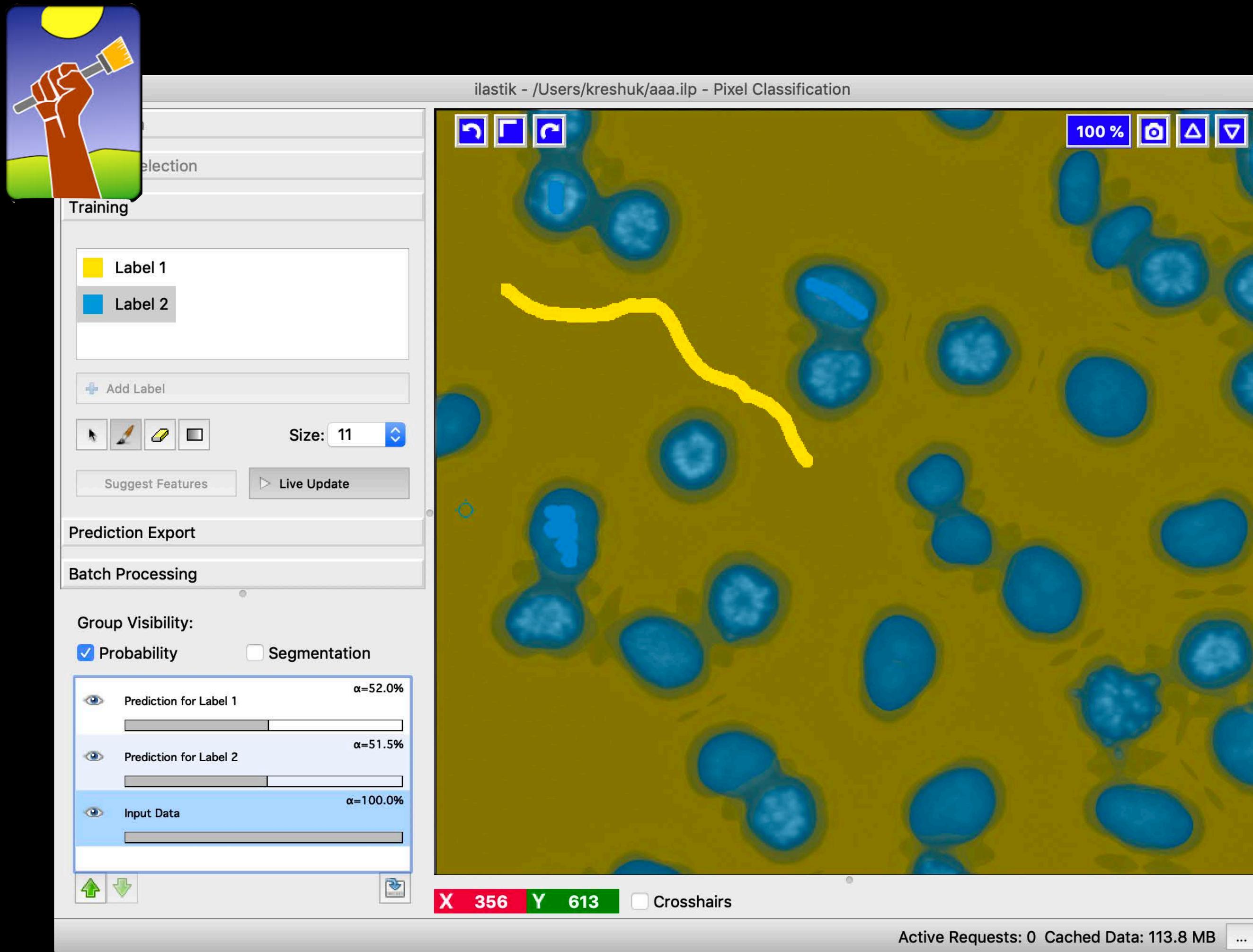
Raw noisy input with denoised and segmentation outputs of a trained DenoISeg Network on Flywing dataset. The network was trained using 1428 noisy images of size 128x128 and only 5 of those images had segmentation ground truth available.

We observe that some errors are present in segmentation result which is to be expected since the network was trained on very limited segmentation annotations. But the errors are significantly lesser as compared to results obtained with other baselines trained with limited ground truth annotations. To compare our results with other baselines, [check this page](#).

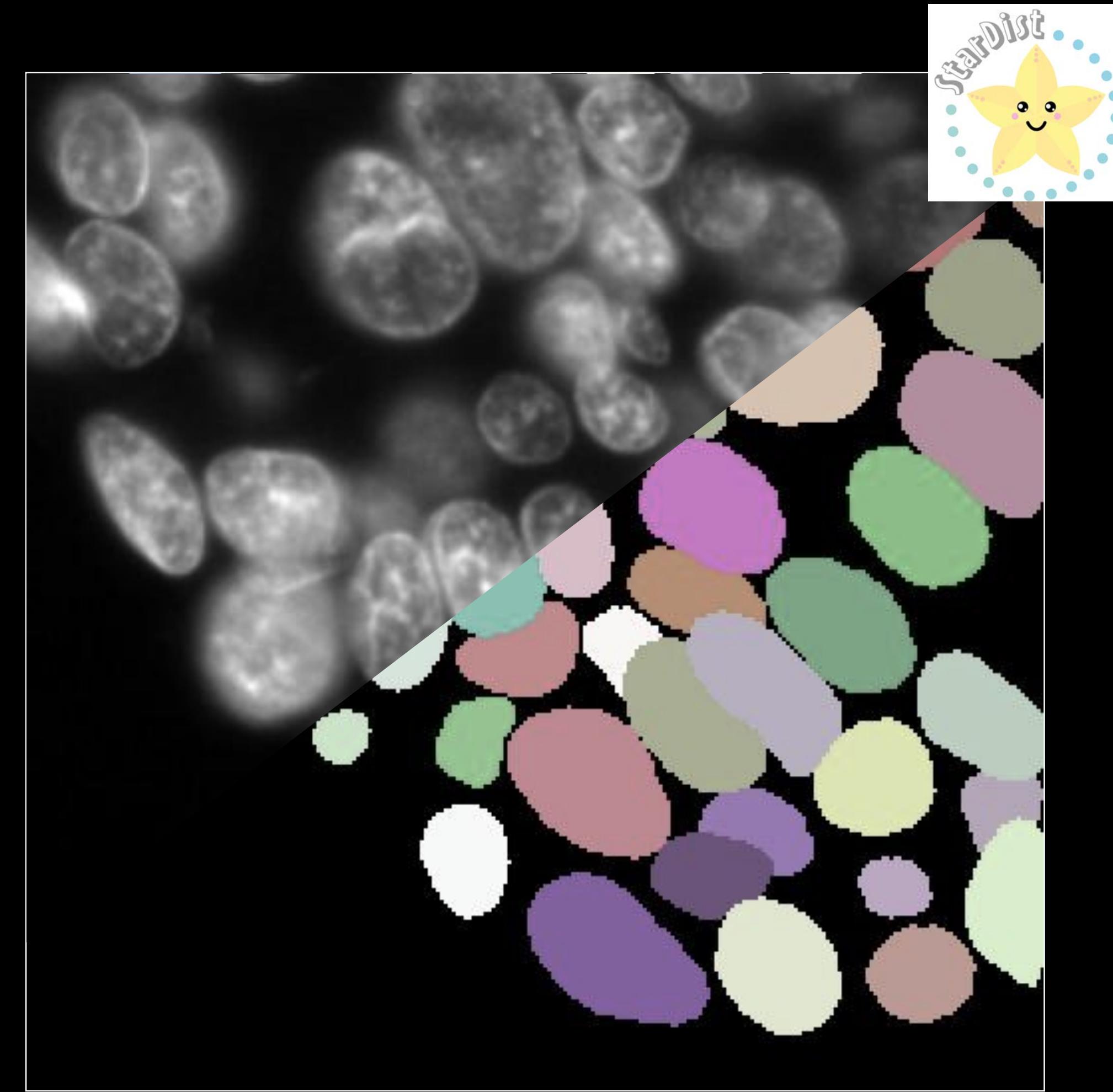
Demo

To reproduce this experiment with our Jupyter notebooks, [follow these instructions](#).

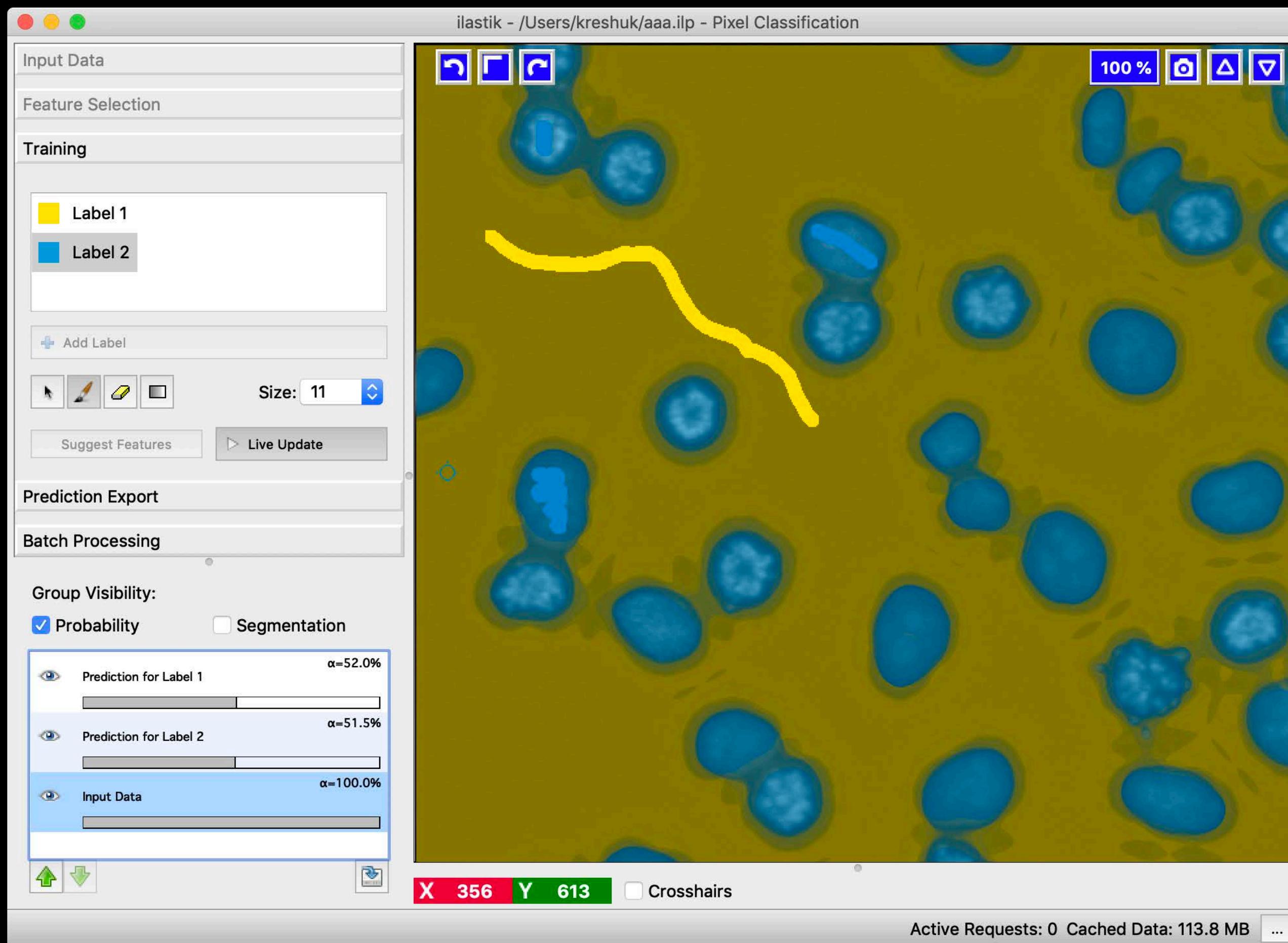




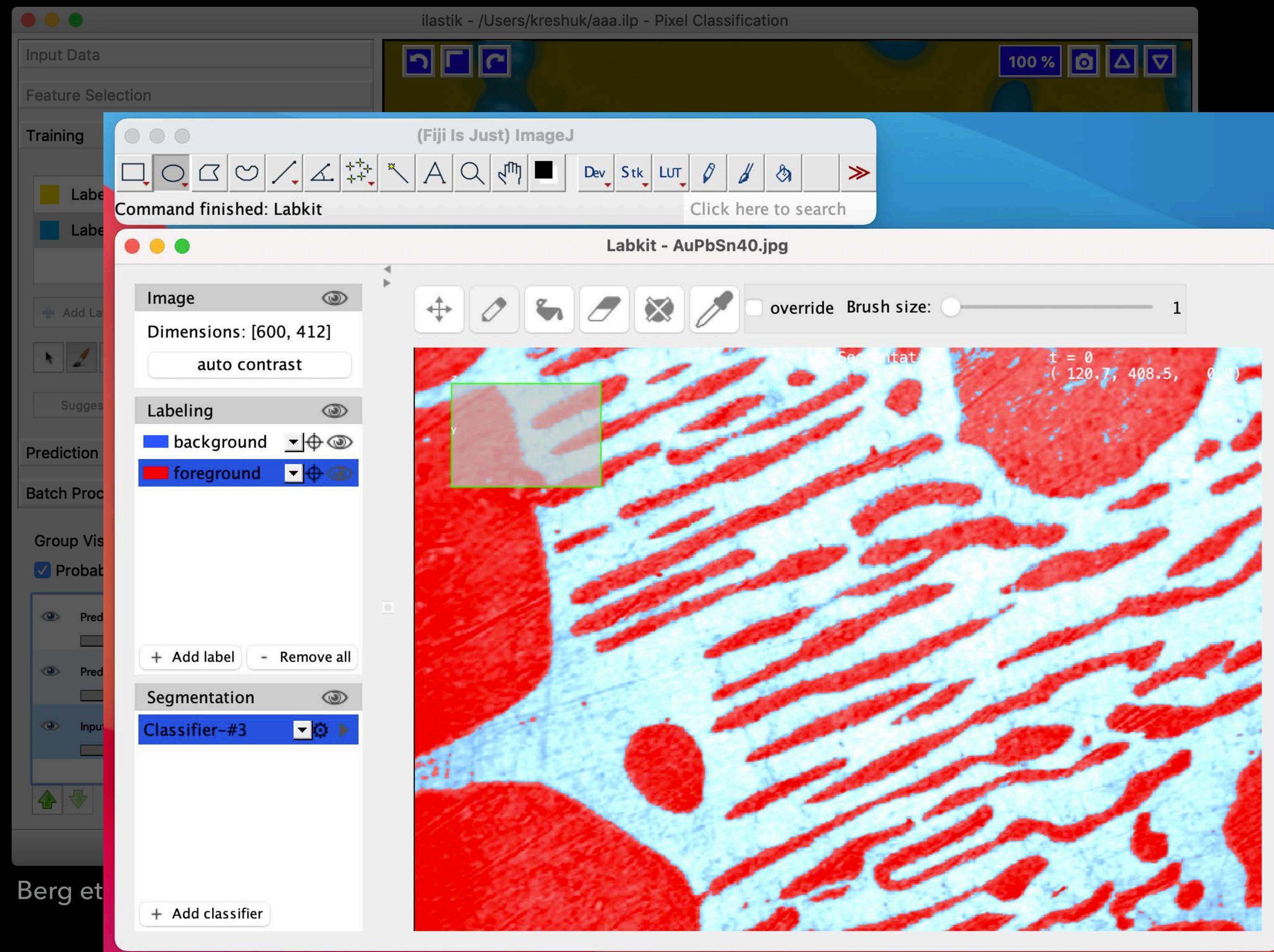
Berg et al., ilastik: interactive machine learning for (bio)image analysis, Nature Methods 2019



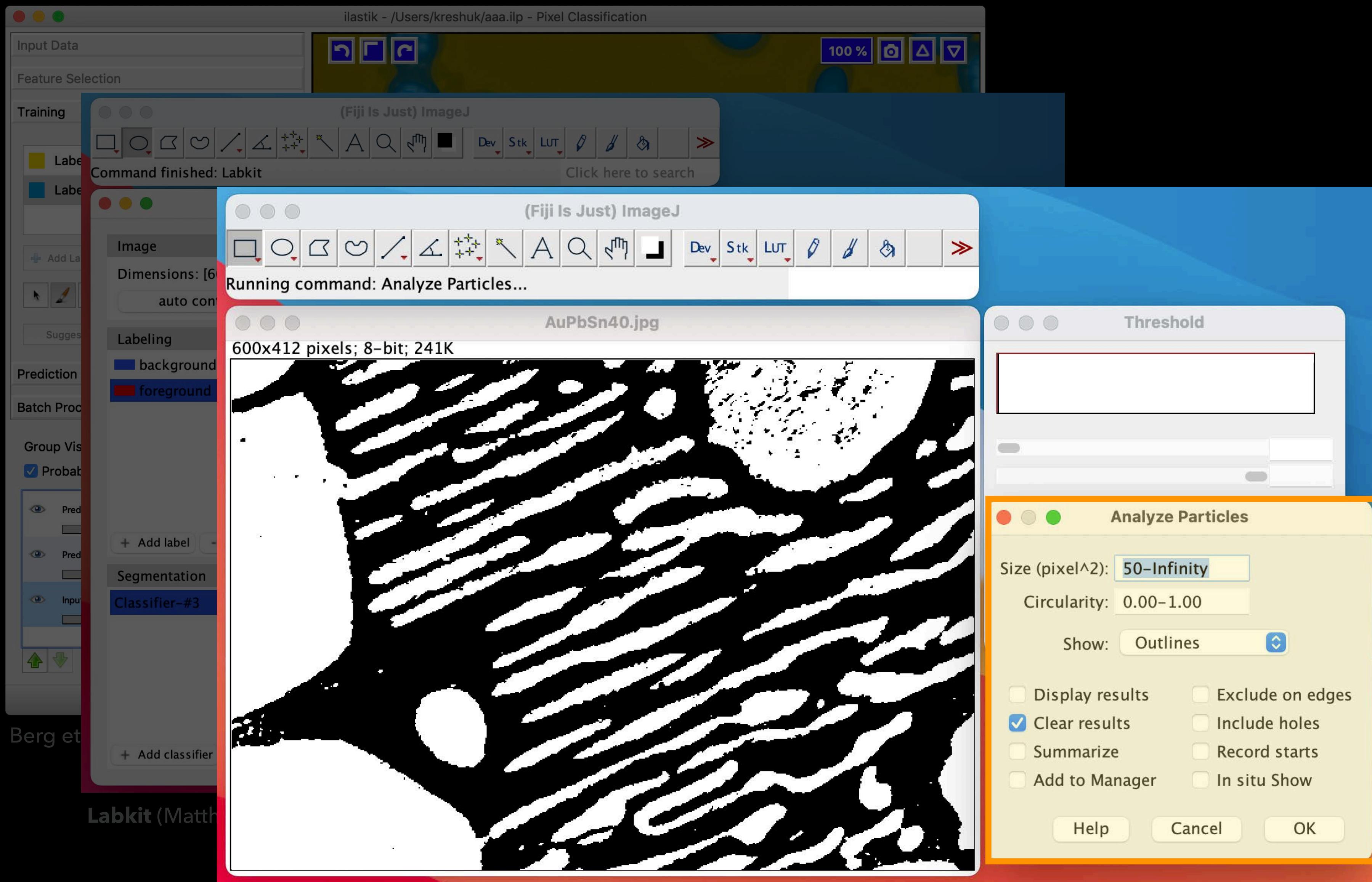
Schmidt et al., Cell Detection with Star-Convex Polygons, MICCAI 2018



Berg et al., ilastik: interactive machine learning for (bio)image analysis, Nature Methods 2019

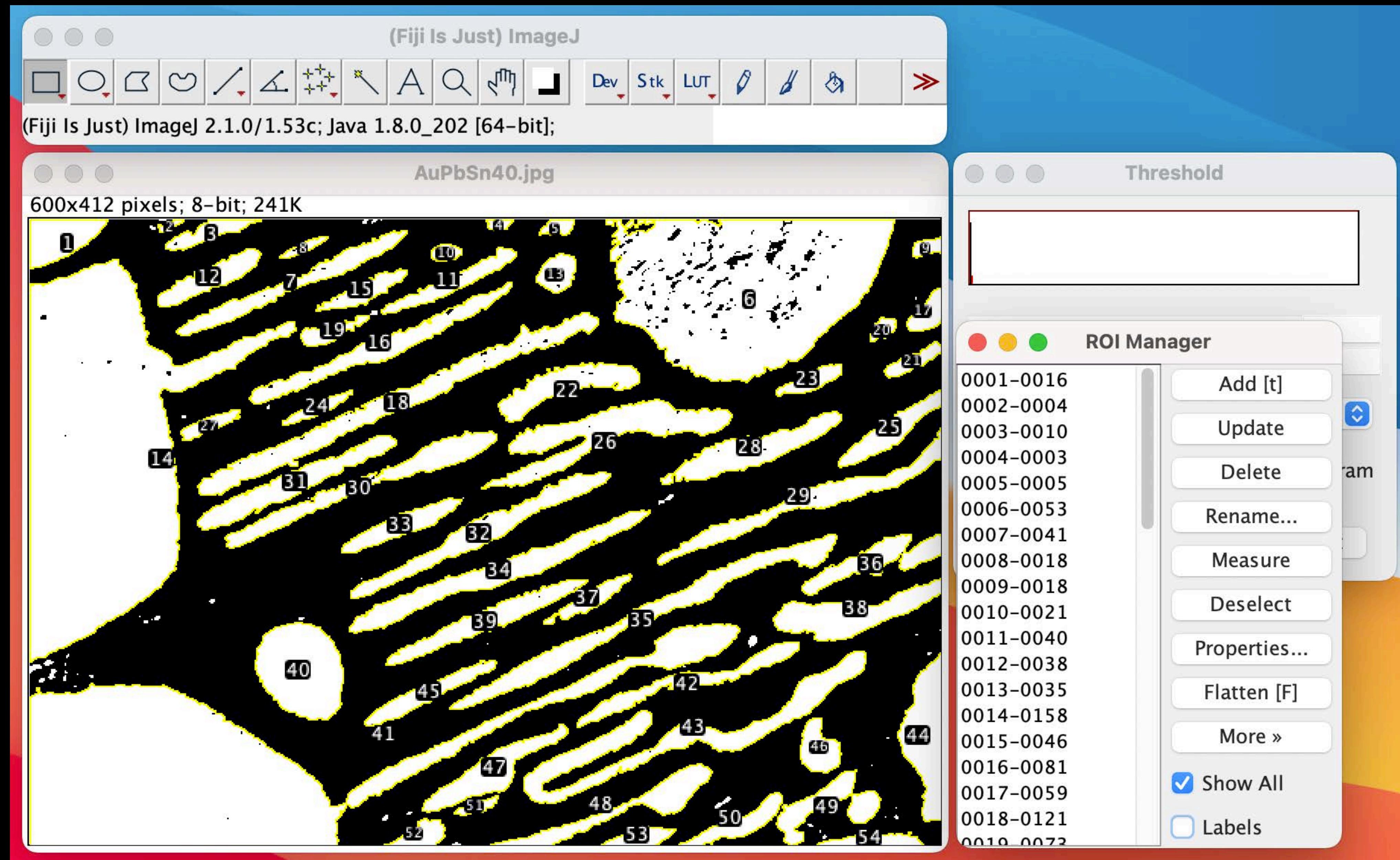


Labkit (Matthias Arzt and others, no publication yet)



so far...
Pixel Classes

to get instances...
**Connected
Component
Analysis**

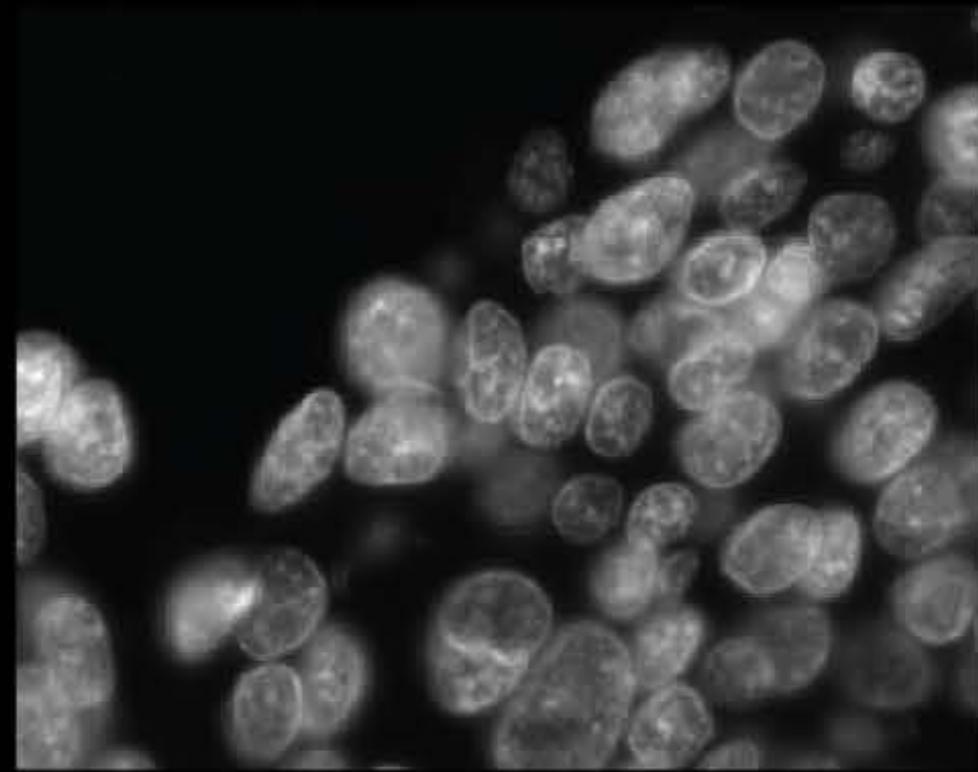


THERE ARE A BUNCH OF
METHODS THAT GIVE YOU
INSTANCES RIGHT AWAY...

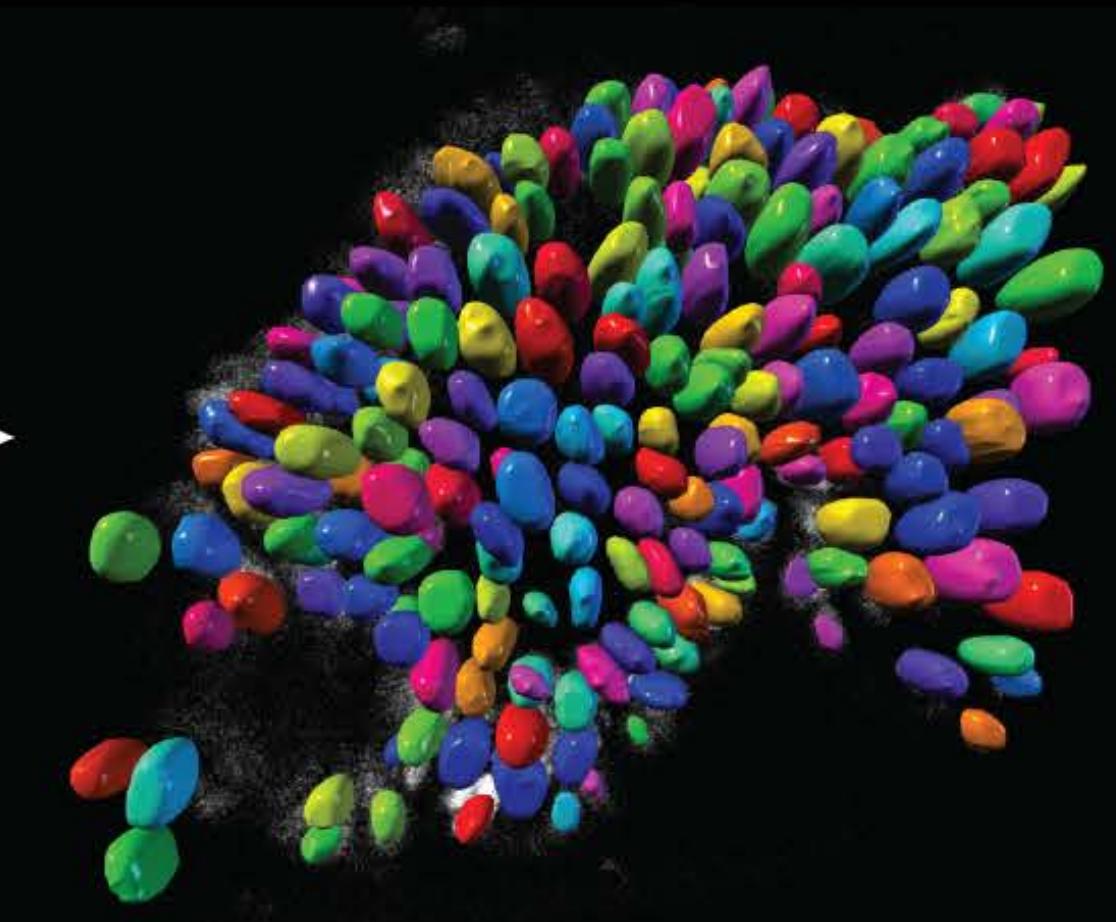
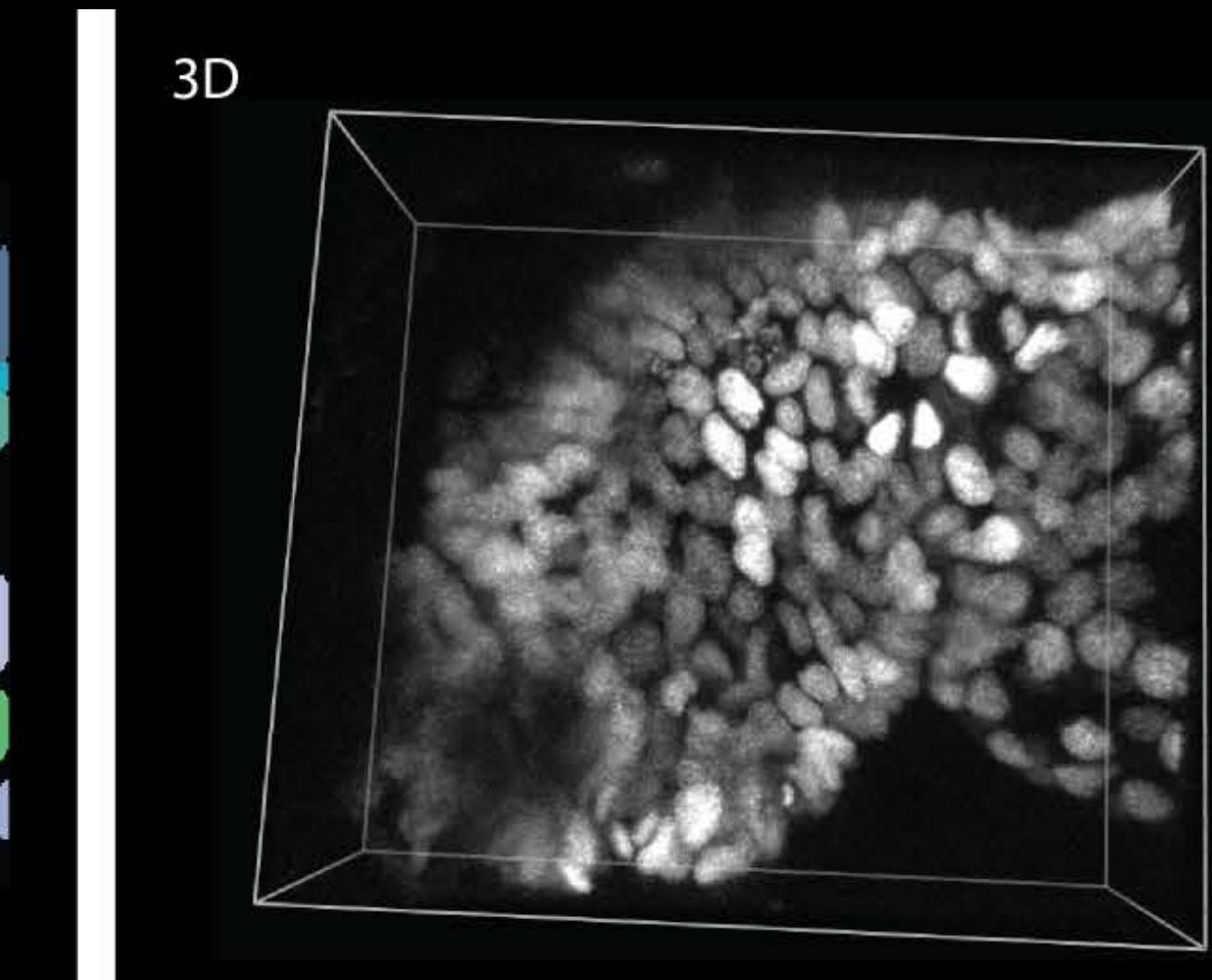
NOT OUR WORK (BUT A *GREAT* METHOD!!!)



2D



3D



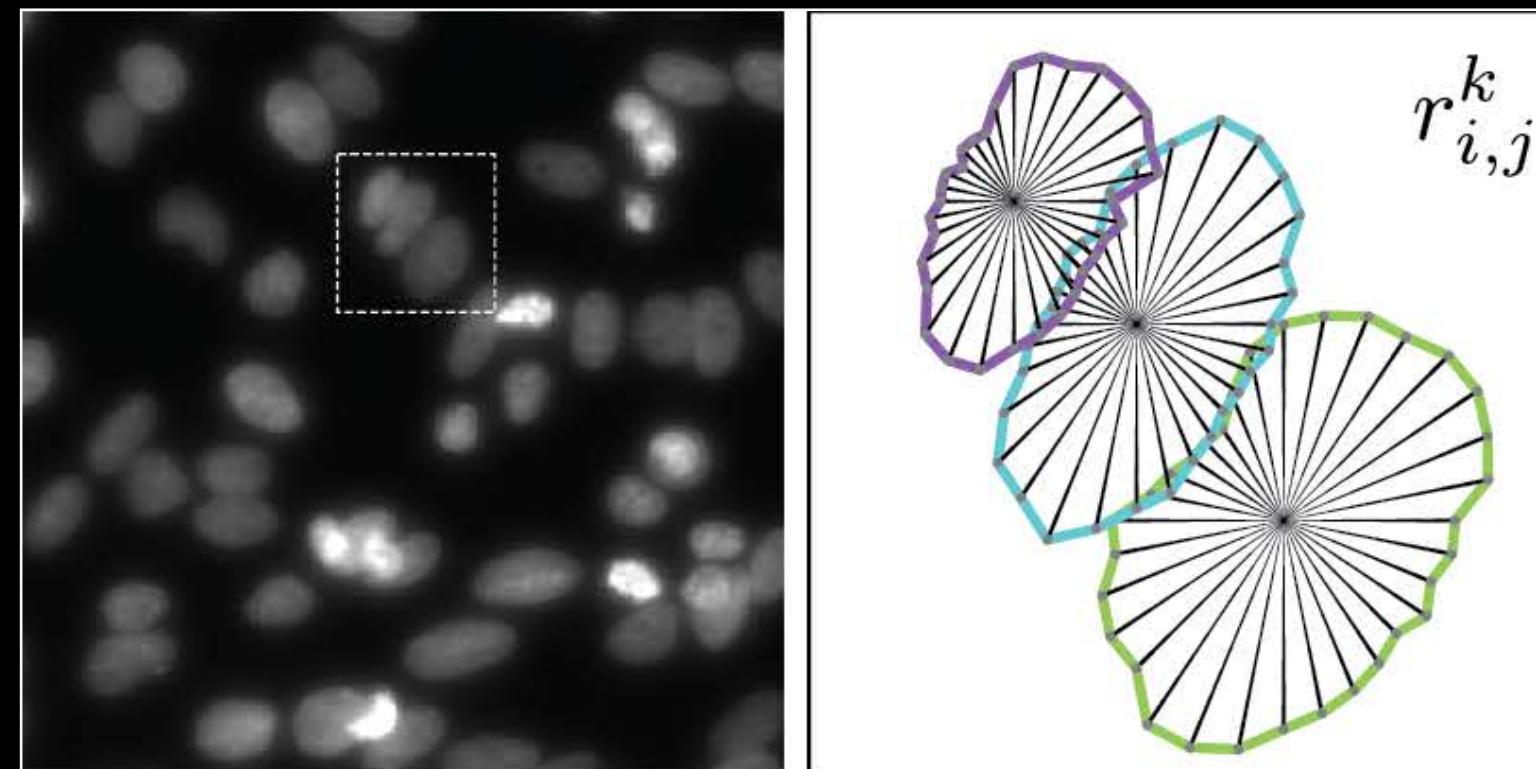
Schmidt et al., Cell Detection with Star-Convex Polygons, MICCAI 2018

Weigert et al., Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy, WACV 2020

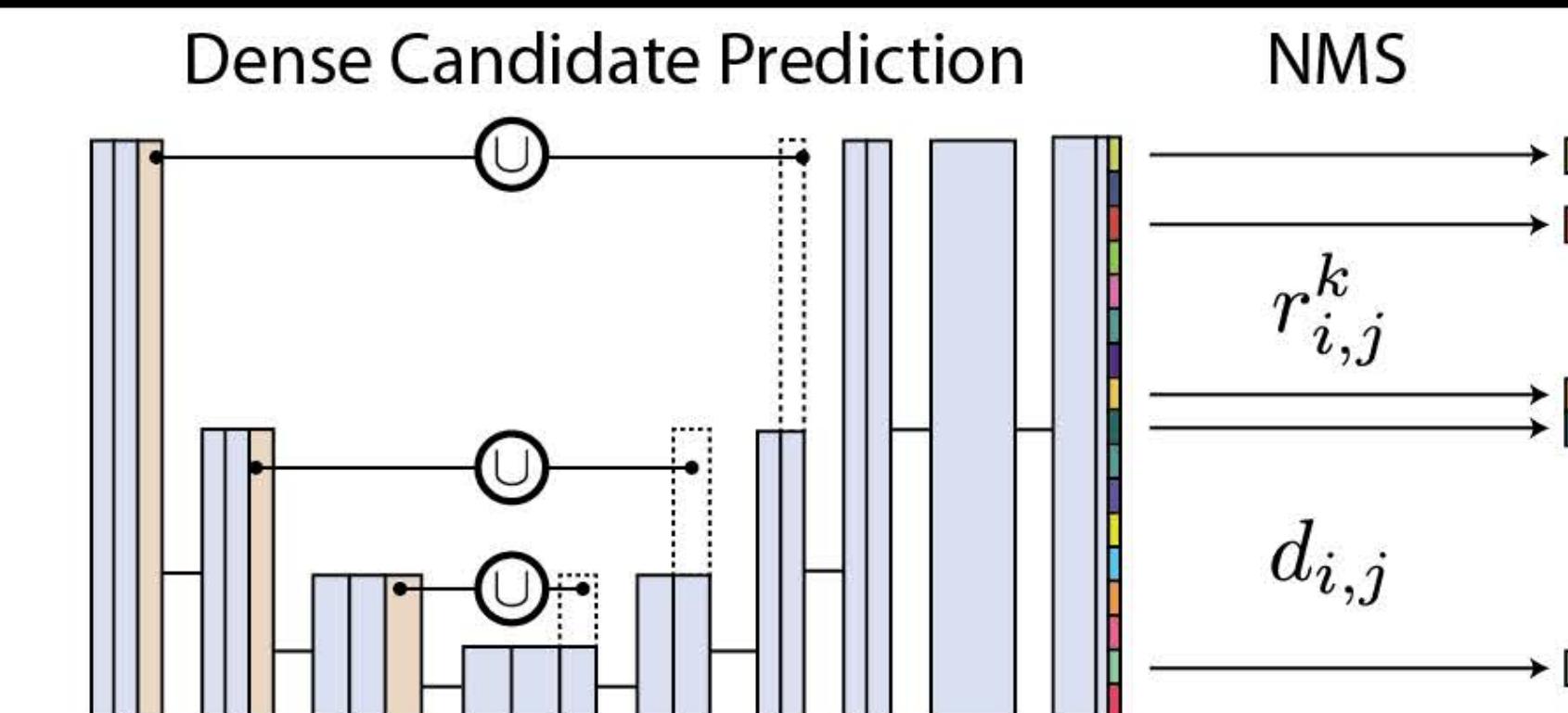
PIXELS VOTE FOR INSTANCE OUTLINE...

NOT OUR WORK (BUT A *GREAT* METHOD!!!)

2D



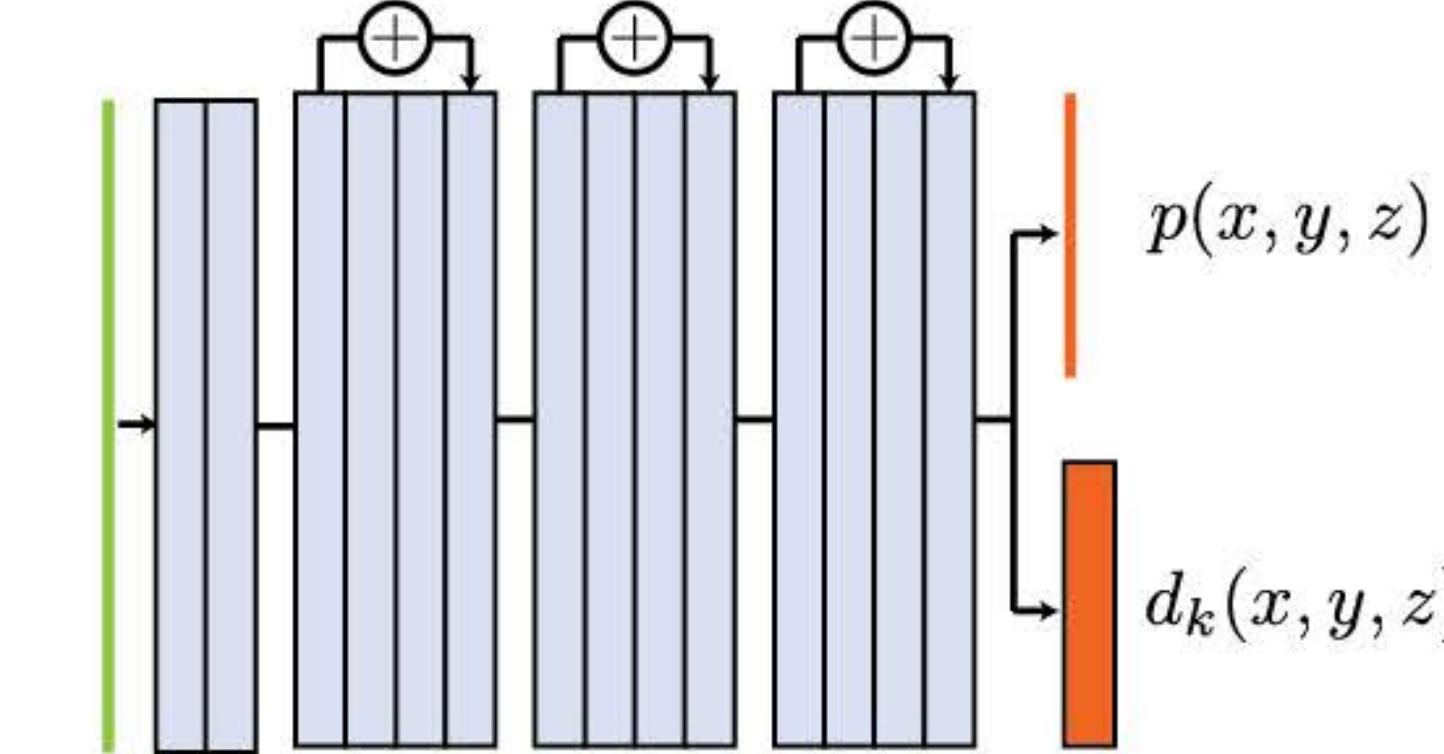
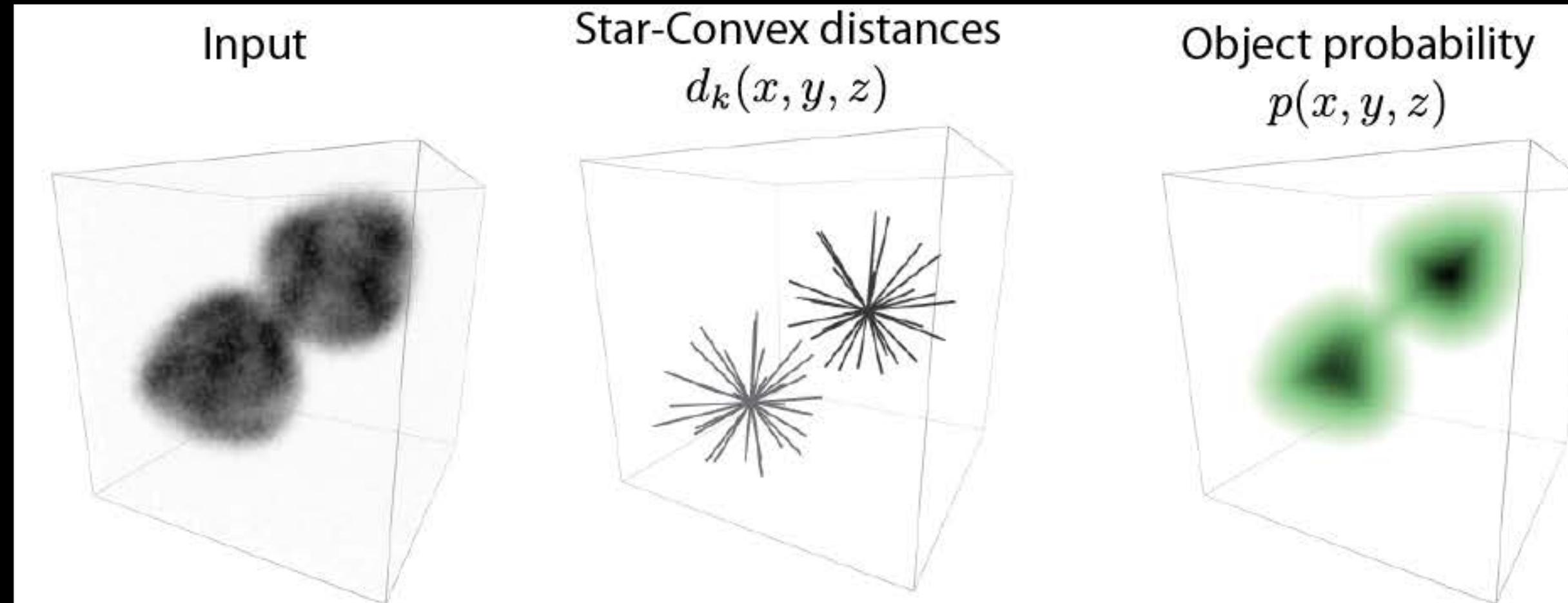
Dense Candidate Prediction

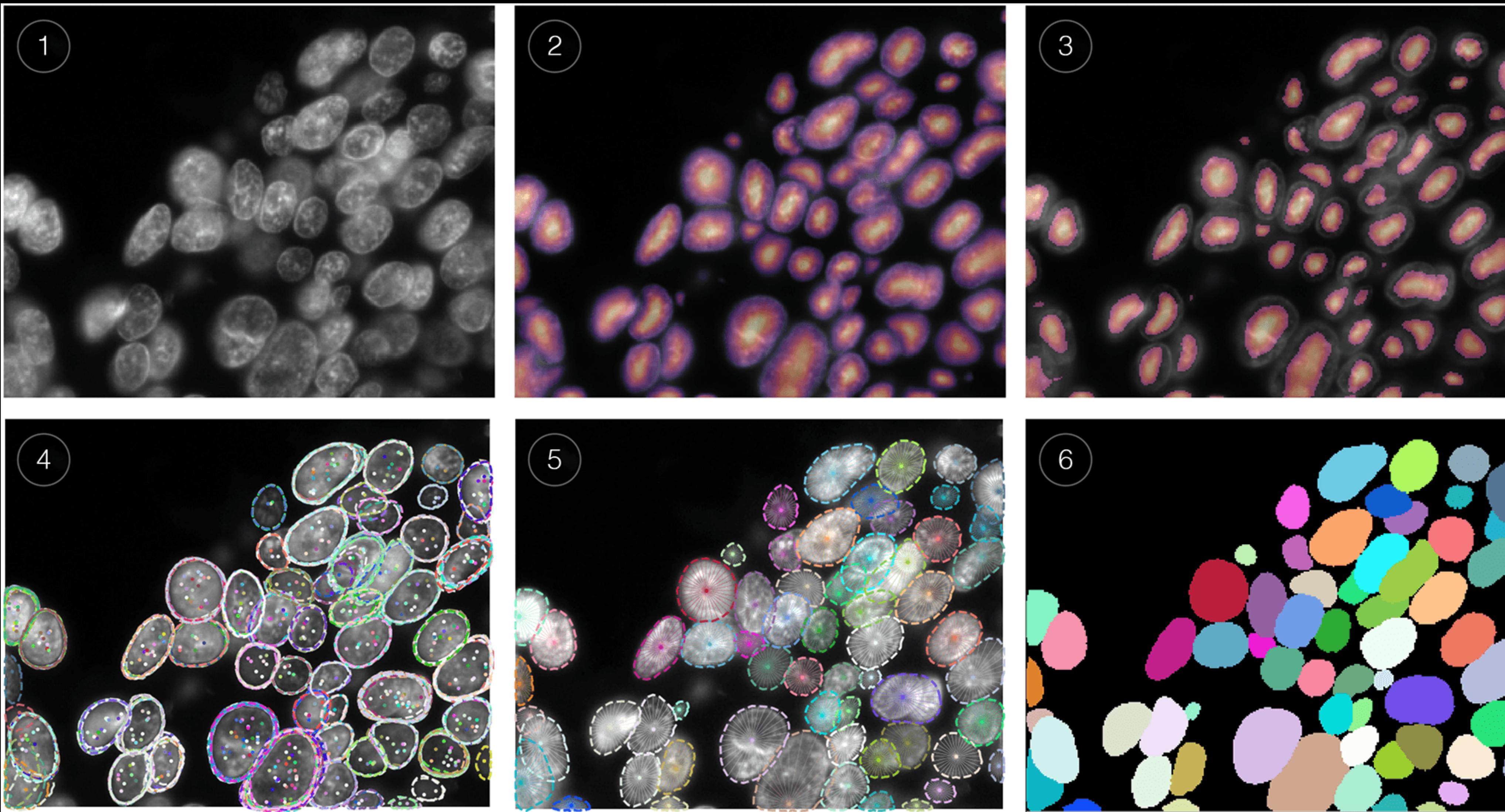


NMS



3D





Schmidt et al., Cell Detection with Star-Convex Polygons, MICCAI 2018

Weigert et al., Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy, WACV 2020

Search or jump to... / Pull requests Issues Marketplace Explore

stardist / stardist Unwatch 21 Star 257 Fork 82

Code Issues 30 Pull requests 1 Actions Projects Wiki Security Insights

master 19 branches 20 tags Go to file Add file Code

uschmidt83 Update README.md 1cd2213 3 hours ago 654 commits

- .github Also build wheels for Python 3.9 7 days ago
- docker Update Dockerfile 5 months ago
- examples Update multiclass example notebook 7 days ago
- extras Merge branch 'master' into dev 20 days ago
- images Tweak image 3 hours ago
- models add 3D demo model 2 years ago
- stardist Revert modelzoo commits (now in own branch) 5 days ago
- tests Revert modelzoo commits (now in own branch) 5 days ago
- .gitignore Update 'other2D' notebooks (#114) 4 months ago
- LICENSE.txt Update year in license last month
- MANIFEST.in Fix MANIFEST.in 7 days ago
- README.md Update README.md 3 hours ago
- pyproject.toml Update pyproject.toml to work around bug on Windows 10 6 months ago
- setup.cfg initial release 3 years ago
- setup.py Support/test Python 3.9 11 days ago

README.md

pypi package 0.7.1 Test passing Test (PyPI) passing forum 94 topics

StarDist - Object Detection with Star-convex Shapes

2D 3D

Search or jump to... / Pull requests Issues Marketplace Explore

HenriquesLab / ZeroCostDL4Mic Watch 27 Star 213 Fork 47

Code Issues 11 Pull requests 10 Actions Projects Wiki Security Insights

Watch 27 Star 213 Fork 47

About

StarDist - Object Detection with Star-convex Shapes

Readme

BSD-3-Clause License

Releases 20

StarDist 0.7.1 Latest 7 days ago + 19 releases

Packages

No packages published

Used by 18

+ 10

Contributors 7

+ 10

Languages

Python	70.7%	C++	19.1%
Jupyter Notebook	6.9%	Groovy	0.7%
C	2.3%		
Makefile	0.2%		
Dockerfile	0.1%		

Home
Romain F. Laine edited this page 20 days ago · 158 revisions

ZeroCostDL4Mic - What is it?

Overview

ZeroCostDL4Mic is a toolbox for the training and implementation of common Deep Learning approaches to microscopy imaging. It exploits the ease-of-use and access to GPU provided by [Google Colab](#).

a) Data upload: Local machine (monitor, server) connects to Google Drive icon.

b) Network training: Google Drive icon connects to Google Colab icon (represented by a yellow triangle).

c) Inference: Google Colab icon connects to Google Drive icon.

d) Data download: Google Drive icon connects back to Local machine.

Notebook, **Training dataset**, **Trained network**, **Dataset**, **Results**, **Data transfer**

Training data can be uploaded to the Google Drive from where it can be used to train models using the provided Colab notebooks in a web-browser. Inference (predictions) on unseen data can then also be performed within the same notebook, therefore not requiring any local hardware or software set-up.

Main:

- Home
- Step by step "How to" guide
- How to contribute
- Tips, tricks and FAQs
- Data augmentation
- Quality control
- Running notebooks locally
- Running notebooks on FloydHub

Fully supported networks:

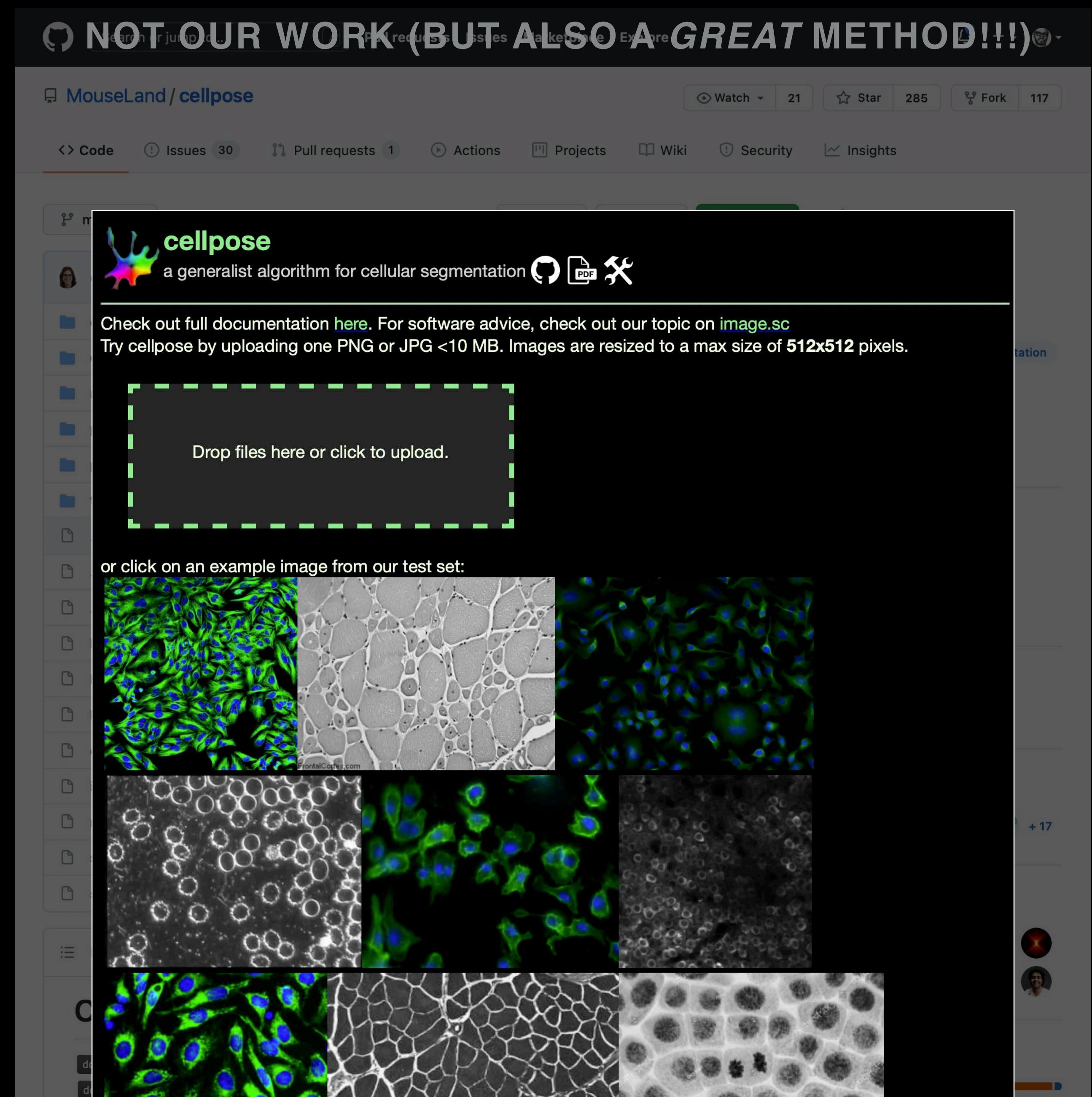
- U-Net
- StarDist
- Noise2Void
- CARE
- Label free prediction (fnet)
- Object Detection (YOLOv2)
- pix2pix
- CycleGAN
- Deep-STORM

Other resources:

- Glossary
- Useful resources
- Example gallery

Clone this wiki locally
<https://github.com/Henric>

Stringer et al., Cellpose: a generalist algorithm for cellular segmentation, Nature Methods 2020



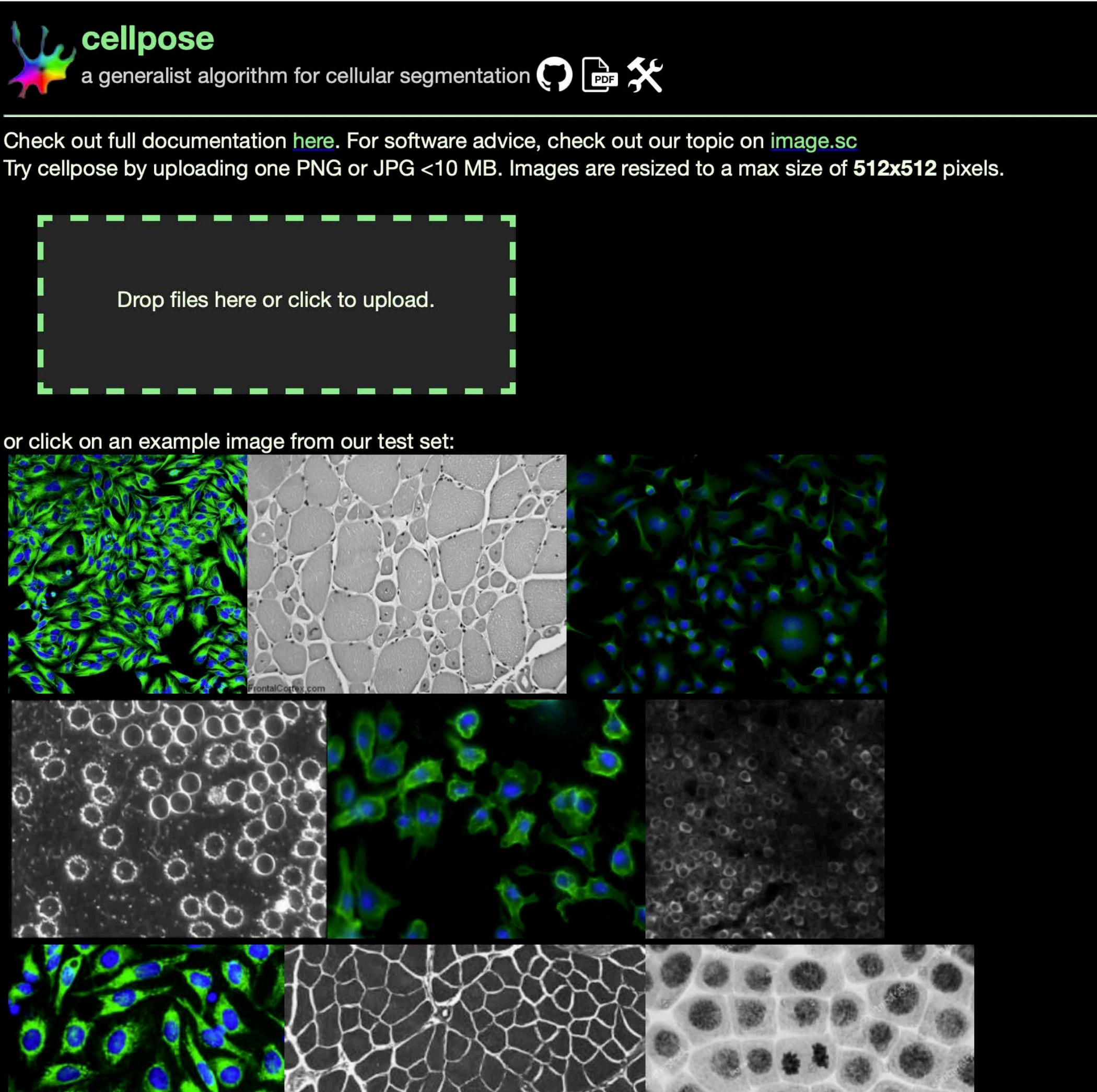
Stringer et al., Cellpose: a generalist algorithm for cellular segmentation, Nature Methods 2020

NOT OUR WORK (BUT ALSO A GREAT METHOD!!!)

MouseLand / [cellpose](#)

[Watch](#) 21 [Star](#) 285 [Fork](#) 117

[Code](#) [Issues](#) 30 [Pull requests](#) 1 [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)



HenriquesLab / [ZeroCostDL4Mic](#)

Search or jump to...

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

[Watch](#) 27 [Star](#) 213 [Fork](#) 47

[Code](#) [Issues](#) 11 [Pull requests](#) 10 [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#)

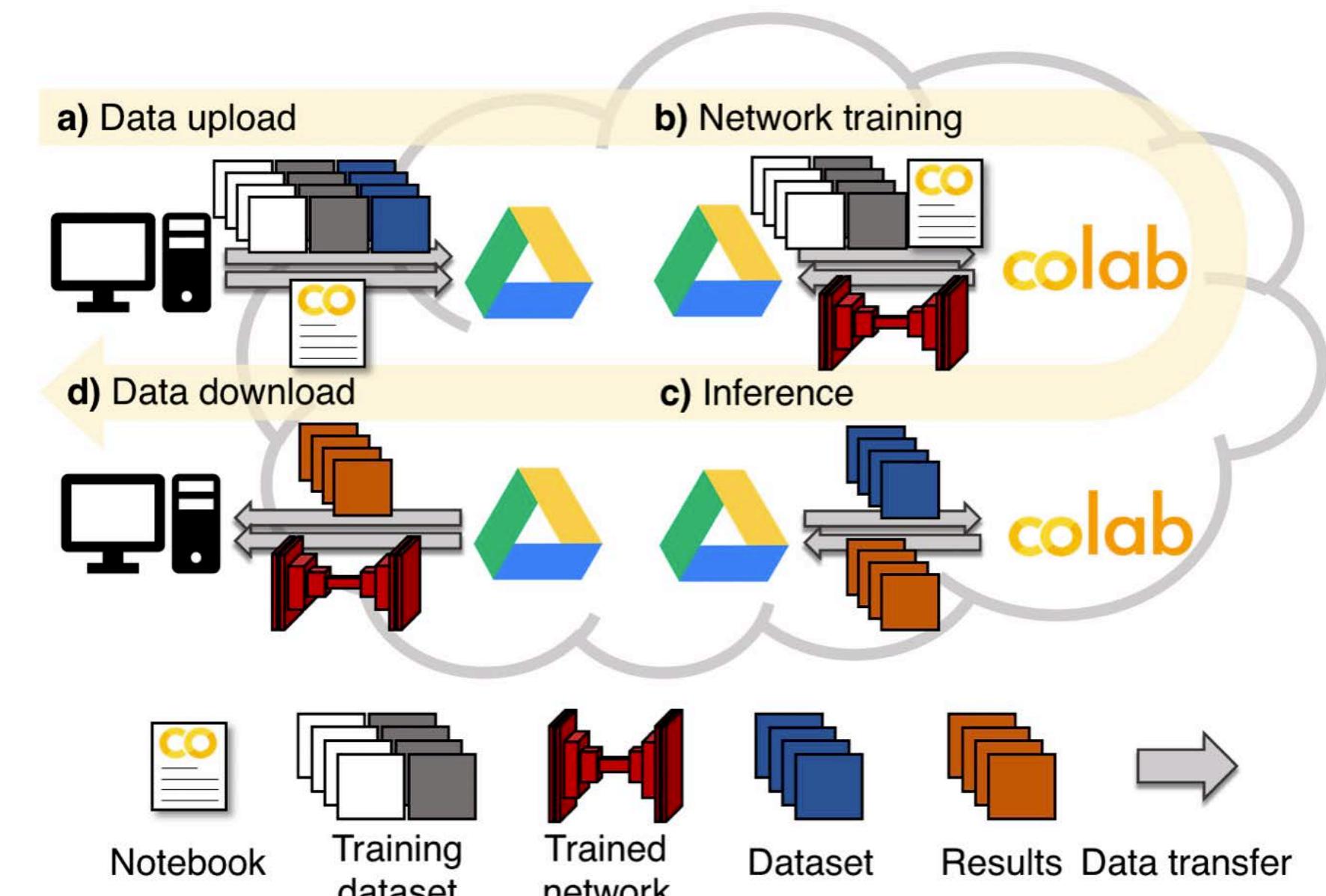
Home

Romain F. Laine edited this page 20 days ago · 158 revisions

ZeroCostDL4Mic - What is it?

Overview

ZeroCostDL4Mic is a toolbox for the training and implementation of common Deep Learning approaches to microscopy imaging. It exploits the ease-of-use and access to GPU provided by [Google Colab](#).



Training data can be uploaded to the Google Drive from where it can be used to train models using the provided Colab notebooks in a web-browser. Inference (predictions) on unseen data can then also be performed within the same notebook, therefore not requiring any local hardware or software set-up.

[Pages 21](#)

Main:

- Home
- Step by step "How to" guide
- How to contribute
- Tips, tricks and FAQs
- Data augmentation
- Quality control
- Running notebooks locally
- Running notebooks on FloydHub

Fully supported networks:

- U-Net
- StarDist
- Noise2Void
- CARE
- Label free prediction (fnet)
- Object Detection (YOLOv2)
- pix2pix
- CycleGAN
- Deep-STORM

Other resources:

- Glossary
- Useful resources
- Example gallery

Clone this wiki locally

<https://github.com/Henric>



Florian Jug
@florianjug

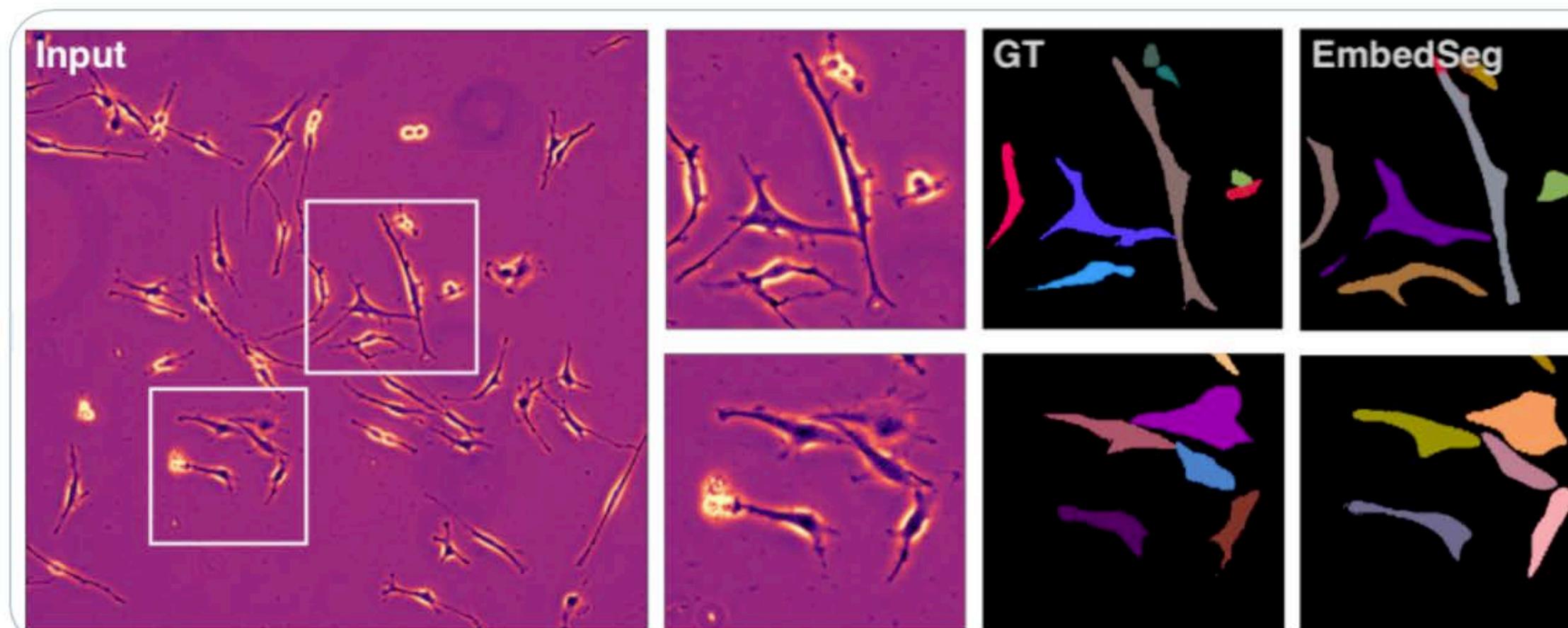
...

Welcome [#EmbedSeg](#), a segmentation method for biomedical data. We show to be competitive with several SOTA baselines and believe to be best-in-class whenever objects have more complex shapes (not being potatoes).

Preprint: arxiv.org/abs/2101.10033

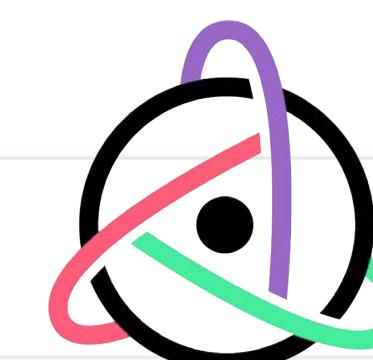
Code: github.com/juglab/EmbedSeg

1/3



12:52 PM · Feb 4, 2021 · Twitter Web App

View Tweet activity

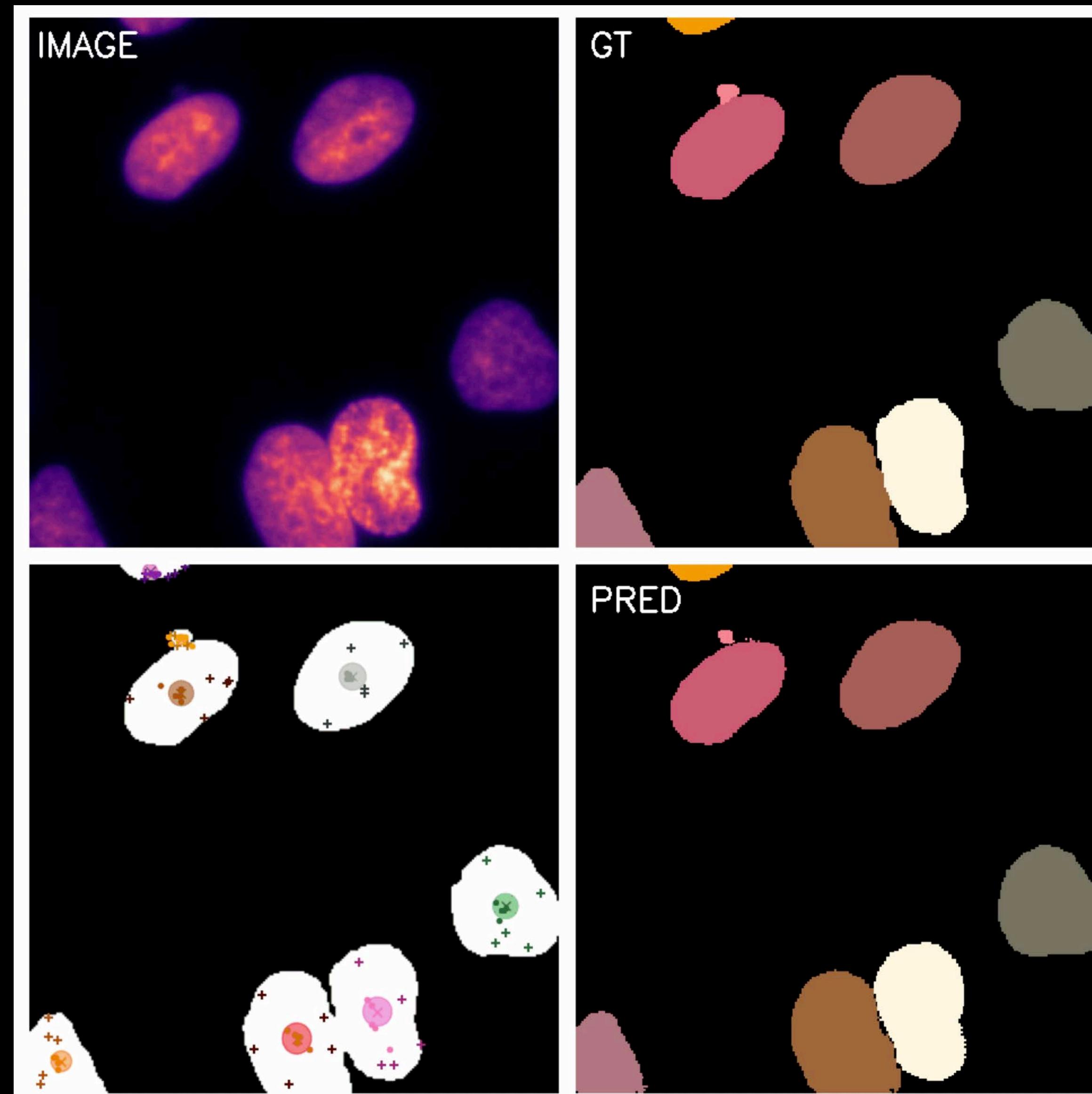


Manan
Lalit

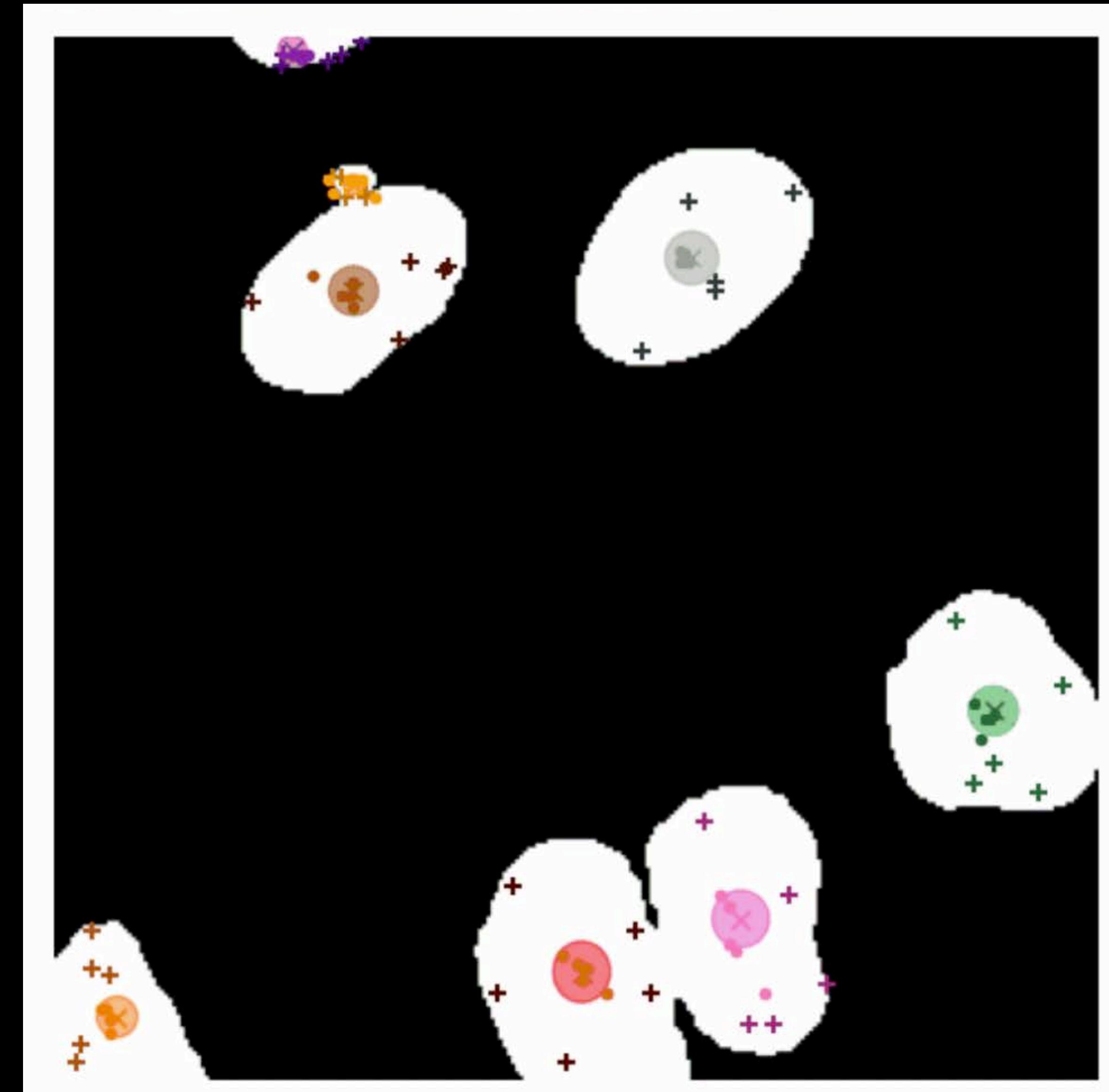


PAvel
Tomancak

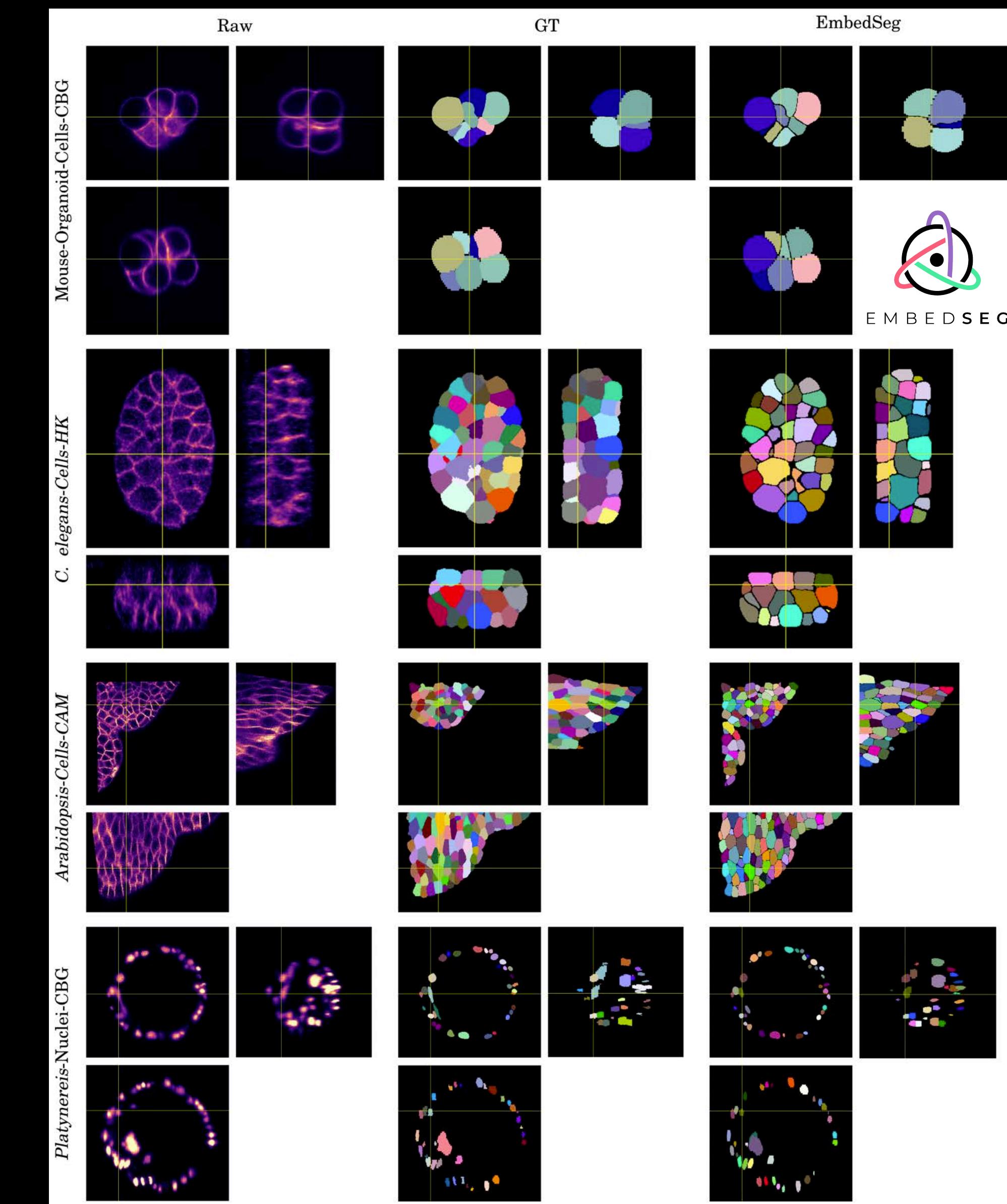
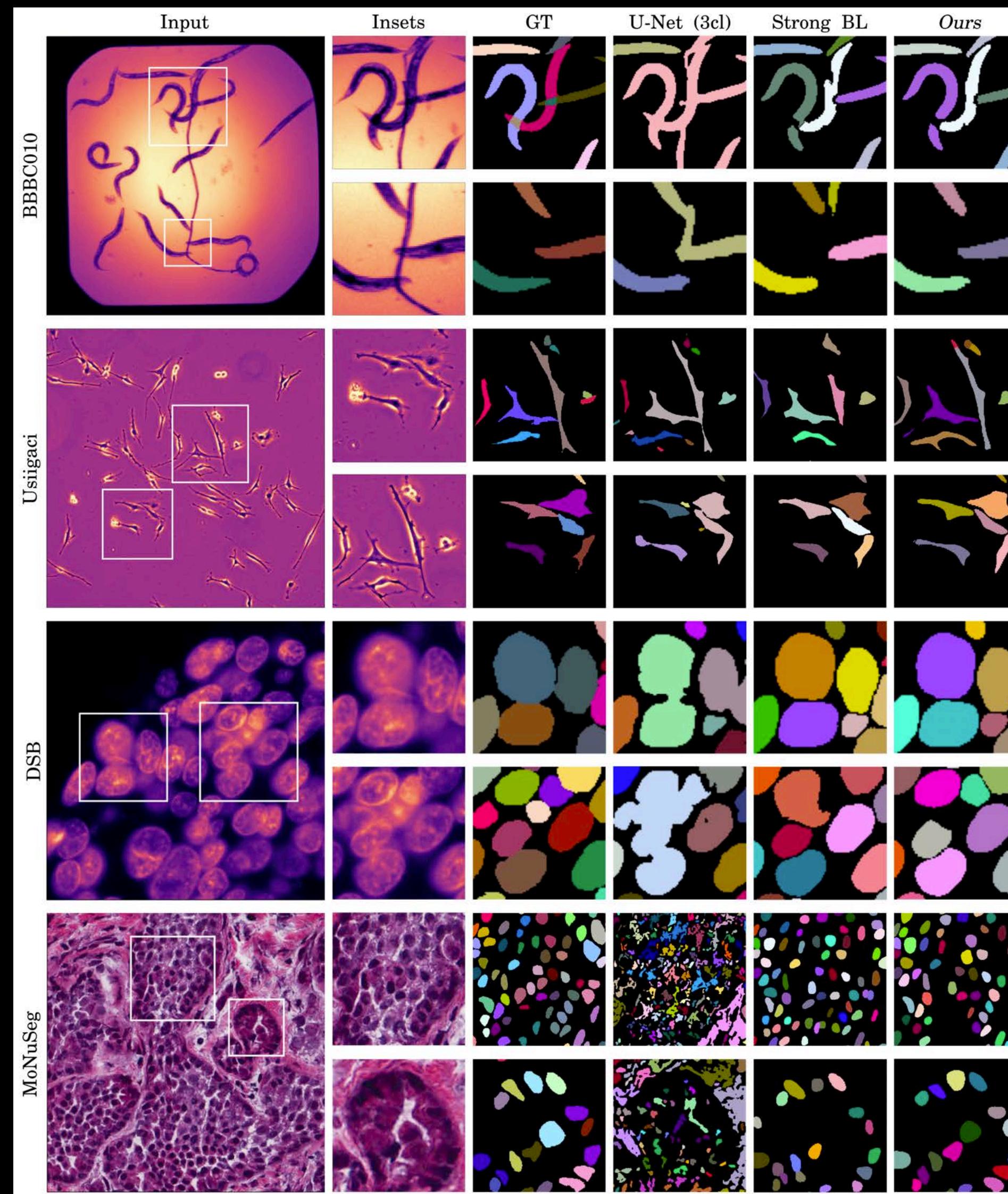
PIXELS VOTE FOR INSTANCE ‘CENTER’...



PIXELS VOTE FOR INSTANCE ‘CENTER’...



WORKS IN 2D AND IN 3D...



WORKS IN 2D AND IN 3D...

	AP _{0.50}	AP _{0.55}	AP _{0.60}	AP _{0.65}	AP _{0.70}	AP _{0.75}	AP _{0.80}	AP _{0.85}	AP _{0.90}
<i>BBBC010</i>									
<i>Usiigaci</i>									
3-Class Unet	0.521	0.466	0.451	0.440	0.427	0.407	0.377	0.332	0.243
Cellpose (<i>public</i>)	0.225	0.204	0.184	0.155	0.097	0.043	0.013	0.002	0.000
Harmonic Emb.	0.900					0.723			
PatchPerPix	0.930		0.905		0.879		0.792		0.386
Neven <i>et al.</i>	0.953	0.941	0.927	0.904	0.878	0.830	0.731	0.563	0.297
EMBEDSEG	0.965	0.954	0.934	0.917	0.896	0.854	0.762	0.596	0.326
<i>DSB</i>									
3-Class Unet	0.806	0.775	0.743	0.701	0.654	0.578	0.491	0.374	0.226
Cellpose (<i>public</i>)	0.868	0.852	0.829	0.802	0.755	0.676	0.563	0.418	0.234
Cellpose (<i>DSB</i>)	0.853	0.826	0.812	0.792	0.768	0.716	0.645	0.536	0.402
Mask R-CNN	0.832	0.805	0.773	0.730	0.684	0.597	0.489	0.353	0.189
PatchPerPix	0.868		0.827		0.755		0.635		0.379
StarDist	0.864	0.836	0.804	0.755	0.685	0.586	0.450	0.287	0.119
Neven <i>et al.</i>	0.873	0.852	0.830	0.799	0.762	0.704	0.623	0.511	0.373
EMBEDSEG	0.876	0.858	0.834	0.806	0.768	0.715	0.645	0.530	0.399
<i>MoNuSeg</i>									
Cellpose (<i>public</i>)	0.757	0.725	0.678	0.610	0.523	0.390	0.222	0.079	0.008
Cellpose (<i>MoNuSeg</i>)	0.726	0.695	0.651	0.597	0.517	0.405	0.256	0.106	<u>0.016</u>
StarDist	<u>0.745</u>	<u>0.709</u>	0.658	0.590	0.491	0.376	0.225	0.088	0.013
Neven <i>et al.</i>	0.704	0.686	0.661	<u>0.618</u>	<u>0.546</u>	<u>0.431</u>	<u>0.274</u>	<u>0.111</u>	<u>0.016</u>
EMBEDSEG	0.717	0.701	0.679	0.636	0.567	0.453	0.294	0.119	0.019

	AP _{0.1}	AP _{0.2}	AP _{0.3}	AP _{0.4}	AP _{0.5}	AP _{0.6}	AP _{0.7}	AP _{0.8}	AP _{0.9}
<i>Mouse-Organoid-Cells-CBG</i>									
Cellpose	0.217	0.214	0.212	0.210	0.203	0.197	0.183	0.146	0.042
StarDist-3D	0.988	0.982	0.982	0.982	0.973	<u>0.970</u>	<u>0.958</u>	<u>0.774</u>	<u>0.052</u>
EMBEDSEG (<i>Full 3D</i>)	0.988	0.982	0.982	0.982	0.973	0.973	0.973	0.970	0.929
<i>C. elegans-Cells-HK</i>									
Cellpose	0.745	0.742	0.740	0.730	0.702	0.673	0.609	0.422	0.013
StarDist-3D	<u>0.959</u>	<u>0.959</u>	<u>0.959</u>	<u>0.959</u>	<u>0.954</u>	<u>0.939</u>	<u>0.881</u>	<u>0.447</u>	<u>0.000</u>
EMBEDSEG (<i>Full 3D</i>)	0.978	0.978	0.978	0.975	0.964	0.948	0.888	0.594	0.013
<i>Arabidopsis-Cells-CAM</i>									
Cellpose	<u>0.266</u>	<u>0.256</u>	<u>0.248</u>	<u>0.241</u>	<u>0.229</u>	<u>0.214</u>	<u>0.196</u>	<u>0.157</u>	<u>0.060</u>
EMBEDSEG (<i>Full 3D</i>)	0.685	0.672	0.661	0.646	0.617	0.584	0.535	0.398	0.100
<i>Platynereis-Nuclei-CBG</i>									
Cellpose	0.971	<u>0.971</u>	<u>0.966</u>	0.957	0.931	0.872	0.700	<u>0.299</u>	0.009
StarDist-3D	<u>0.973</u>	<u>0.969</u>	<u>0.966</u>	<u>0.966</u>	<u>0.937</u>	<u>0.910</u>	<u>0.736</u>	<u>0.246</u>	<u>0.002</u>
EMBEDSEG (<i>Full 3D</i>)	0.982	0.982	0.982	0.975	0.964	0.932	0.804	0.361	<u>0.004</u>
<i>Paryhale-Nuclei-IGFL</i>									
U-Net	<u>0.592</u>	0.552	0.481	0.372	0.280	0.198	0.097	0.010	0.000
Cellpose	<u>0.545</u>	0.498	0.456	0.384	0.285	0.154	0.040	0.006	0.000
StarDist-3D	0.766	0.757	0.741	0.698	0.593	0.443	0.224	0.038	<u>0.000</u>
EMBEDSEG	0.581	<u>0.581</u>	<u>0.579</u>	<u>0.543</u>	<u>0.472</u>	<u>0.359</u>	<u>0.185</u>	0.038	<u>0.000</u>
<i>Mouse-Skull-Nuclei-CBG</i>									
Cellpose	0.613	0.587	0.587	0.563	<u>0.515</u>	<u>0.471</u>	<u>0.389</u>	<u>0.316</u>	<u>0.064</u>
StarDist-3D	0.468	0.468	0.400	0.358	<u>0.264</u>	<u>0.138</u>	<u>0.034</u>	<u>0.000</u>	<u>0.000</u>
EMBEDSEG (<i>Sliced</i>)	<u>0.649</u>	<u>0.649</u>	<u>0.649</u>	<u>0.649</u>	0.424	0.424	0.237	0.146	0.108
EMBEDSEG (<i>Full 3D</i>)	0.837	0.837	0.837	0.837	0.795	0.646	0.549	0.362	0.053
<i>Platynereis-ISH-Nuclei-CBG</i>									
Cellpose	0.731	0.674	0.629	0.554	0.493	0.390	0.247	0.038	0.000
StarDist-3D	0.599	0.587	0.545	0.442	0.280	0.114	0.010	0.000	0.000
EMBEDSEG (<i>Sliced</i>)	0.893	<u>0.872</u>	<u>0.831</u>	<u>0.821</u>	<u>0.745</u>	<u>0.634</u>	0.509	0.175	<u>0.000</u>
EMBEDSEG (<i>Full 3D</i>)	<u>0.884</u>	0.884	0.874	0.852	0.781	0.655	<u>0.482</u>	<u>0.120</u>	<u>0.000</u>

>> Lalit, Tomancak, Jug 2021 (MIDL)

GITHUB

>> Lalit, Tomancak, Jug 2021 (MIDL)

 [juglab / EmbedSeg](#)

Code Issues Pull requests Actions Projects Wiki Security Insights

main 3 branches 2 tags Go to file Add file Code

 [MLbyML Merge pull request #5 from juglab/v0.1](#) ... cb93b79 7 days ago 73 commits

EmbedSeg Comment import of `hdmedians` package 7 days ago

cmaps Add base functionality last month

examples Add sentence about using `Labkit` to prepare instanc... 11 days ago

pretrained_models Add `pretrained_models` for all example datasets 18 days ago

.gitignore Update `.gitignore` file to not consider `2d` directory 18 days ago

EmbedSeg_environment.y... Re-export `environment.yml` file 19 days ago

README.md Correct path to `examples` directory 8 days ago

license.txt Add license file last month

setup.py Add setup.py file last month

README.md

EmbedSeg

Introduction

This repository hosts the version of the code used for the [preprint](#) **Embedding-based Instance Segmentation of Microscopy Images**. For a short summary of the main attributes of the publication, please check out the [project webpage](#).

We refer to the techniques elaborated in the publication, here as **EmbedSeg**. **EmbedSeg** is a method to perform instance-segmentation of objects in microscopy images, based on the ideas by [Neven et al, 2019](#).

About

Code Implementation for the preprint "Embedding-based Instance Segmentation for Microscopy Images"

[juglab.github.io/embedseg/](#)

deep-learning
instance-segmentation
test-time-augmentation
spatial-embeddings

Readme
View license

Releases

2 tags

Create a new release

Packages

No packages published

[Publish your first package](#)

Contributors 2

 [MLbyML](#) Manan Lalit

 [fjug](#) Florian Jug



GITHUB

[juglab / EmbedSeg](#)

Code Implementation for the preprint "Embedding-based Instance Segmentation for Microscopy Images"

[MLbyML Merge pull request #5 from juglab/v0.1](#) · cb93b79 · 7 days ago · 73 commits

[EmbedSeg](#) · Comment import of `hdmedians` package · 7 days ago

[cmaps](#) · Add base functionality · last month

[examples](#) · Add sentence about using 'Labkit' to prepare instance... · 11 days ago

[pretrained_models](#) · Add `pretrained_models` for all example datasets · 18 days ago

[.gitignore](#) · Update `.gitignore` file to not consider `2d` directory · 18 days ago

[EmbedSeg_environment.y...](#) · Re-export `environment.yml` file · 19 days ago

[README.md](#) · Correct path to `examples` directory · 8 days ago

[license.txt](#) · Add license file · last month

[setup.py](#) · Add setup.py file · last month

[README.md](#)

EmbedSeg

Introduction

This repository hosts the version of the code used for the [preprint](#) Embedding-based Instance Segmentation of Microscopy Images. For a short summary of the main attributes of the publication, please check out the [project webpage](#).

We refer to the techniques elaborated in the publication, here as **EmbedSeg**. EmbedSeg is a method to perform instance-segmentation of objects in microscopy images, based on the ideas by [Neven et al, 2019](#).

No packages published
Publish your first package

Contributors 2

- MLbyML Manan Lalit
- fjug Florian Jug

**JUPYTER NOTEBOOKS**

[localhost:8888/notebooks/02-train.ipynb](#)

13 Begin training!

Executing the next cell would begin the training.

If `display` attribute was set to `True` above, then you would see the network predictions at every n^{th} step (equals 5, by default) on training and validation images.

Going clockwise from top-left is:

- * the raw-image which needs to be segmented,
- * the corresponding ground truth instance mask,
- * the network predicted instance mask, and
- * (if `display_embedding = True`) from each object instance, 5 pixels are randomly selected (indicated with `+`), their embeddings are re plotted (indicated with `.`) and the predicted margin for that object is visualized as an axis-aligned ellipse centred on the ground-truth - center (indicated with `x`) for that object

```
In [ ]: begin_training(train_dataset_dict, val_dataset_dict, model_dict, loss_dict)
```

Common causes for errors during training, may include :

1. Not having center Images for both train and val directories
2. Mismatch between type of center-images saved in `01-data.ipynb` and the type of center chosen in this notebook (see the `center` parameter in the third code cell in this notebook)
3. In case of resuming training from a previous checkpoint, please ensure that the model weights are read from the correct directory, using the `resume_path` parameter. Additionally, please ensure that the `save_dir` parameter for saving the model weights points to a relevant directory.

[localhost:8888/notebooks/03-predict.ipynb](#)

4 Begin Evaluating

Setting `verbose` to True shows you Average Precision at IOU threshold specified by `ap_val` above for each individual image. The higher this score is, the better the network has learnt to perform instance segmentation on these unseen images.

```
In [11]: #matplotlib agg
begin_evaluating(test_configs, verbose = False, avg_bg = avg_bg/normalize)

2-D 'test' dataloader created! Accessing data from ../../data/db-2018/test/
Number of images in 'test' directory is 50
Number of instances in 'test' directory is 50
Number of center images in 'test' directory is 0
*****
Creating branched erfnet with [4, 1] classes
100% [50/50] [01:00<00:00, 1.22s/it]
Mean Average Precision at IOU threshold = 0.5, is equal to 0.87935
```

Common causes for a low score/error is:

1. Accessing the model weights at the wrong location (this could happen, for example, if you access the prediction on a day different from the one when you trained the notebook). Simply editing the `checkpoint_path` would fix the issue.
2. At times, you would notice an improved performance by lowering `seed_thresh` from 0.90 (default) to say 0.80.
3. CUDA error: out of memory - ensure that you shutdown `02-train.ipynb` notebook before running this notebook.

NAPARI PLUGIN

Specify path to crops and center embedding

Data Directory [Browse](#)
Center [medoid](#)

Obtain properties of the dataset

Data Properties [Browse](#)

Specify training dataset related parameters

Train Size 1200
Train Batch Size 1
Train Virtual Batch Size 1

Specify validation dataset related parameters

Val Size 800
Val Batch Size 1
Val Virtual Batch Size 1

Specify additional parameters

Number of Epochs 200
 Display ? Display Embedding ?
Save Directory [Browse](#)
Resume Path [Browse](#)

Begin training

Begin training Stop training
Epoch
Train Loss 13%
Val Loss

EmbedSeg

Specify path to the evaluation images

Data Directory [Browse](#)

Specify path to the model weights and data properties

Path to model weights [Browse](#)
Path to data properties [Browse](#)

Specify evaluation parameters

Test Time Augmentation ?
Average Precision Label 0.5
Seediness Threshold 0.90
 Save Images ? Save Results ?
Path to Results [Browse](#)

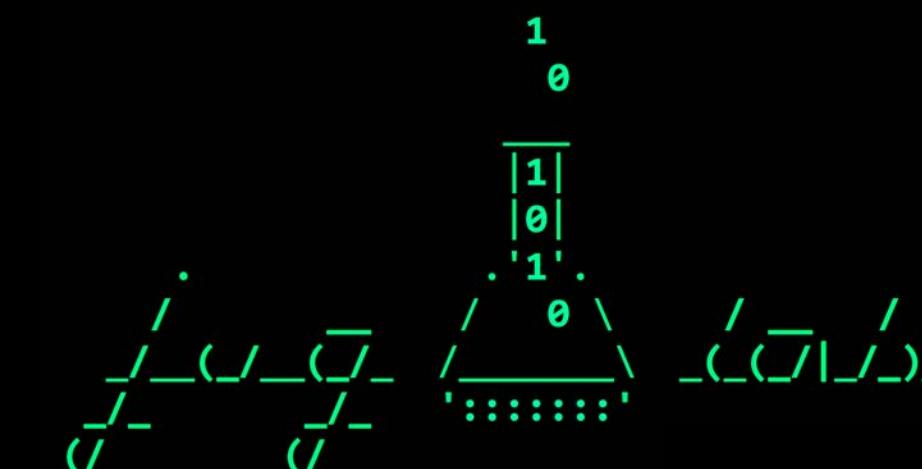
Begin evaluating

Predict 32%

>> Lalit, Tomancak, Jug 2021 (MIDL)



Alex Krull
Alumnus Postdoc



**Tim-Oliver
Buchholz**
Alumnus PhD st.



**Mangal
Prakash**
Alumnus PhD st.



**Tobias
Pietzsch**
Superhero



**Anna
Goncharova**
PhD Student



**Deborah
Schmidt**
Alumnus RSE



**Christopher
Schmied**
Bioimage Analyst



**Tom
Burke**
RSE



**Matthias
Arzt**
RSE



**Uwe
Schmidt**
Freigeist



**Joran
Deschamps**
RSE



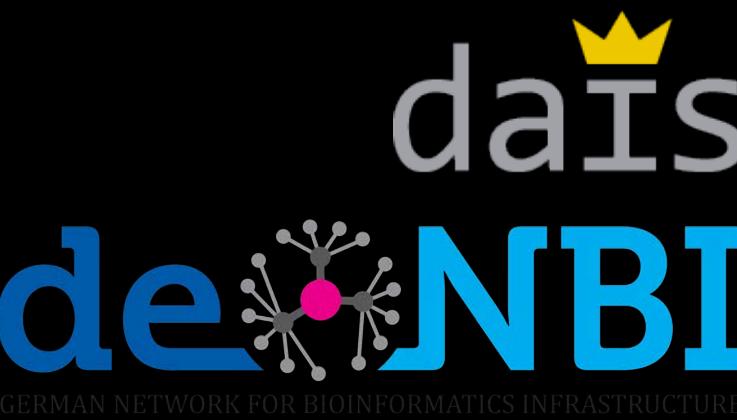
**Nuno
Pimpao M.**
PhD Student



**Manan
Lalit**
PhD Student



**Martin
Weigert**
Wunderkind



THANKS - DANKE - GRAZIE - MERCI - GRACIAS - DĚKUJI - ASANTE - HVALA

Model Zoo Collab- orators

Euro-BioImaging ERIC
Henriques Lab (IGC)
Jug Lab (CSBD/HT)
Kreshuk Lab (EBML-HD)
Lundberg Lab (KTH)
deepImageJ Team
Biolmage Archive (EMBL EBI)
Partner LS Ris

